NPS55-79-019

NAVAL POSTGRADUATE SCHOOL

Monterey, California

FACPUB: A SYSTEM FOR COMPUTERIZING

FACULTY PUBLICATION RECORDS

by

Gilbert T. Howard

September 1979

Prepared for:
Naval Postgraduate School
Monterey, Ca 93940

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

Rear Admiral T. F. Dedman
Superintendent

J. R. Borsting
Provost

This report was prepared by

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS55-79-019	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: FACPUB: A System for Computerizing Faculty Publication Records.	5. TYPE OF REPORT & PERIOD COVERED: Technical
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Gilbert T. Howard	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93940	12. REPORT DATE: September 1979
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Publication Records
Faculty Publications
Information and Retreival
Naval Postgraduate School

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes the FACPUB system for the computerization of faculty publication records at the Naval Postgraduate School. The system includes interactive APL programs for record entry and correction and FORTRAN programs for printing. The report contains program descriptions and listings.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS

TABLE OF CONTENTS Cont.

TABLE OF CONTENTS Cont.

TABLE OF CONTENTS Cont.

I. INTRODUCTION AND ACKNOWLEDGMENTS

The Faculty Publication System, FACPUB, a system to computerize the faculty publication records of the NPS faculty, was created in response to several demands. The Dean of Research recognized the need for a compilation of the publications and presentations of the faculty. The Research Office periodically issues a report, *"Recent NPS Publications,"* [1] covering the most recent three years and a computerized listing would assist in its preparation. In addition the NPS library is asked to respond to requests for lists of technical reports, books and other publications produced by NPS faculty.

The system was developed under the urging of Dean W. M. Tolles by the author while serving as the Assistant to the Dean of Research. Generous help and most of the APL programs in the system were provided in their initial form by Professor K. T. Marshall who developed them for use in publishing the OR/MS Index [2]). Numerous changes have been made in these programs, but the FACPUB system would never have existed in its current form without them.

Ms. Helen J. Waldron and Ms. Julie Diepenbrock Herrick of the Cataloging Division of the NPS Library were involved in initial system specifications and Ms. Herrick has been heavily involved in overseeing the actual data input and in suggesting improvements in the programs used for that purpose. The burden of data input has so far fallen largely on Ms. Rumi Esocbido of the Cataloging Division although substantial help was provided in the early phase of the project by Mrs. Linda Ishii.

Ms. Patricia Meadows of the Operations Research Department deserves recognition for having written the programs LIST, BOOK, GETSORT and GETBOOK.

II. SYSTEM OVERVIEW

A. System Requirements

The basic requirement for the FACPUB system was that it provide a convenient way of entering, sorting, and printing faculty publication records. More specifically the following requirements were identified:

1. The input process should be convenient for the operator and should not require that the operator be a skilled computer user.
2. The system must be capable of producing upper and lower case output.
3. The system must be capable of sorting records according to the following characteristics:
   a. academic department
   b. publication type
   c. faculty member
   d. security classification
   e. year of publication
   f. alphabetically by first author.

It was never intended that the system should be an on-line system for answering inquiries about specific author's records. Such inquiries will be answered by reference to master lists of publications produced periodically.

B. System Design

The FACPUB system is a terminal oriented system as far as the file construction is concerned. Records are entered by a user working interactively with APL programs operating under CMS. The records are written on private disk space and periodically written onto tape for back-up protection and as necessary

when the available disk space is filled. The records are printed from the tape using FORTRAN programs operating under OS. Sorting can be done conveniently in CMS, but larger files must be sorted under OS and sorting program GETSORT was prepared for this purpose. It is also envisioned that once the files are constructed for a given year and written on tape that GETSORT and the accompanying print program LIST will be used to sort and print these files producing master copies for the library's use.

Figure II.1 shows the system design.

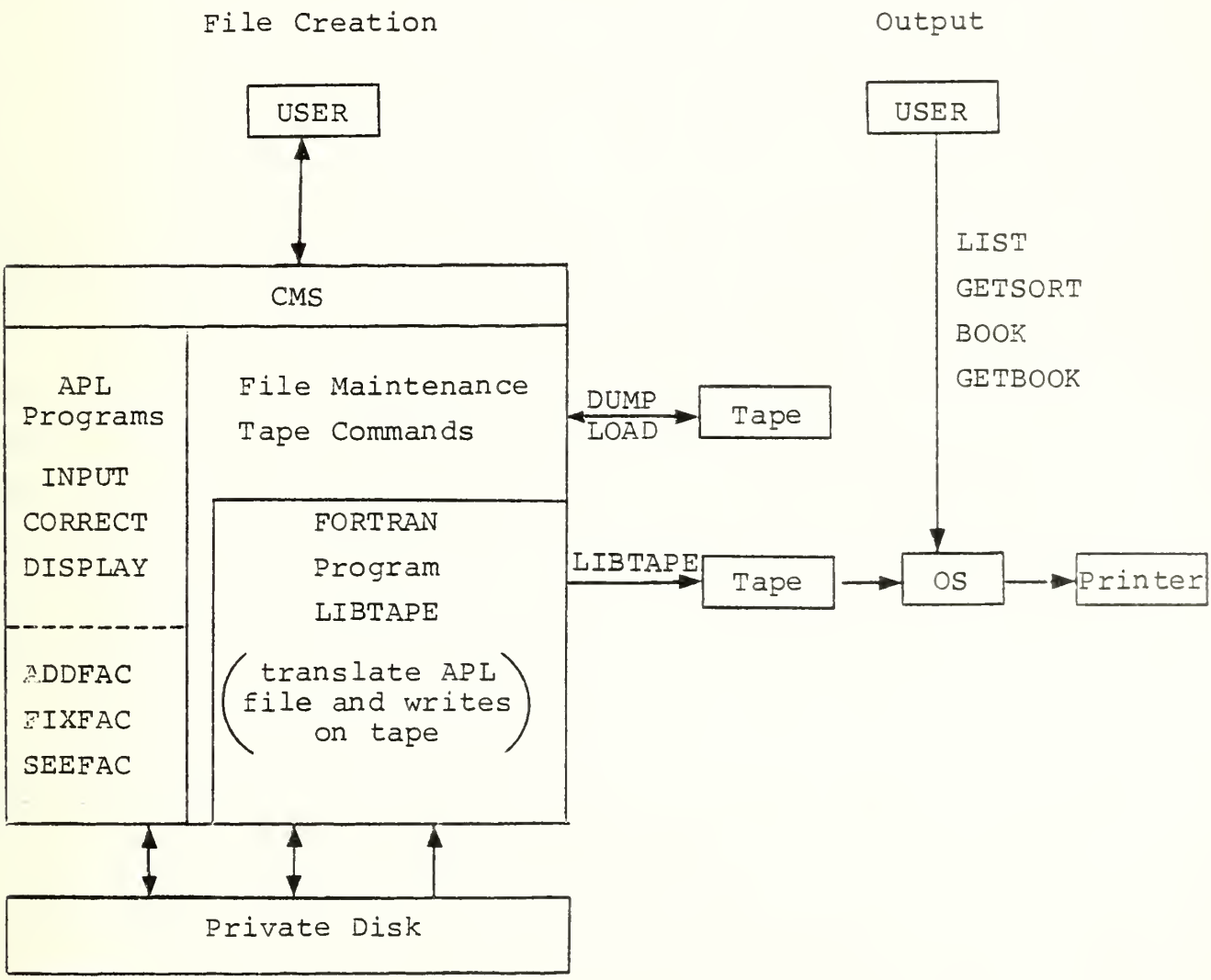


Figure II.1

C. Input Data for FACPUB

The input data for the FACPUB system is organized in the library's records by individual faculty member. The primary source of this information is through the annual Faculty Activity Reports. Obtaining accurate and complete data in a timely manner has been a major problem to the library. In fact a side effect of using the FACPUB system has been to identify a significant discrepancy between the records made available to the library and those in the academic departments. Previous issues of the volume entitled *"Recent NPS Publications"* contain numerous entries not forwarded to the library through the Faculty Activity Reports. In 1978 a change was made in the form of these reports with the intention of soliciting more complete data for the library, but it is too early to tell if it has been successful.

Another problem in the input data is that multiple authors will often report the same publication differently. Perhaps the titles will vary, or the citations might differ, or more commonly, the authors permute the names.

Since the input data is organized by individual faculty, duplicate entries for multiple authors are expected. A significant manual effort would have to be made to avoid this duplication and it was not felt to be worthwhile. Provision has been made within FACPUB to identify primary and secondary listings (see Section III-A3 for details) but for this to be successful it is necessary that duplicate entries (for multiple authors) all list the same first author.

Another minor problem in the data is that some faculty members hold joint appointments between departments, and some have changed departments. This causes some ambiguity with respect to the department code assigned to each entry, but it is still possible to sort out any individual's publications by his faculty ID number.

III. DETAILED SYSTEM DESCRIPTION

A. Record Format

Each entry in the system is stored in a record 800 characters long. The records are written by INPUT which is an APL program operating under CMS. Table III.1 is a description of the contents of each record. Following the table is a further description of the contents of each field.

Field No.	Start	End	Length	Contents
1	1	3	3	Department Code
2	4	4	1	Publication Code
3	5	5	1	Print Code
4	6	8	3	Faculty ID Number
5	9	9	1	Security Classification
6	10	12	3	Year of Publication
7	13	13	1	Number of Authors
8	14	43	30	Author Number 1
9	44	73	30	Author Number 2
10	74	103	30	Author Number 3
11	104	133	30	Author Number 4
12	134	163	30	Author Number 5
13	164	193	30	Author Number 6
14	194	223	30	Author Number 7
15	224	253	30	Author Number 8
16	254	493	240	Title
17	494	736	243	Citation
18	737	800	64	Translated Authors

TABLE III.1. Record Contents

1. Department code

Up to three digits identifying the department code associated with the author for whom this publication is being entered. For faculty with joint appointments and faculty who have changed departments this is ambiguous, but probably the primary department or the current department should be entered. Field 4 allows a sort by faculty member so that all publications of a specified author can be listed.

2. Publication code

A one-character code which identifies the type of publication according to the following scheme. The INPUT program accepts the library's two- or three-character code (in upper case) and translates it to the single-character form for storage in the record.

LIBRARY'S CODE	ONE CHARACTER CODE	MEANING
UNP	A	Unpublished
BK	B	Book
TBK	C	Textbook
PRC	D	Contributed Paper, not published
PRI	E	Invited Paper, not published
SYC	F	Contributed Paper, published in Proceedings
SYI	G	Invited Paper, published in Proceedings
SYE	H	Editor of Symposium Proceedings
LTR	I	Letter to editor
TN	J	Technical Note
PP	K	Paper Published in Journal
PI	L	Invited Paper in open literature
ABC	M	Abstract of Contributed Paper
ABI	N	Abstract of Invited paper
CHB	O	Chapter of book

LIBRARY'S CODE	ONE CHARACTER CODE	MEANING
RPR	P	Project Report
RPT	Q	NPS Technical Report
PAT	R	Patent application or award
REV	S	Book Review
TR	T	Translation
THE	U	Thesis
MIS	V	Miscellaneous, pamphlets, etc.

3. Print code

A single digit used in the printing programs. Its primary function is to suppress the printing of duplicate entries. Recall that the input is organized by individual faculty member and publications with multiple NPS authors will be entered more than once. The print code is assigned automatically by the INPUT program and has the following meaning:

0 The faculty ID number entered in field 4 cannot be found in the list of faculty ID's. (The ID number should be checked or the faculty directory should be updated to include this author).

1 Primary listing. The faculty ID number of this record corresponds to the first author.

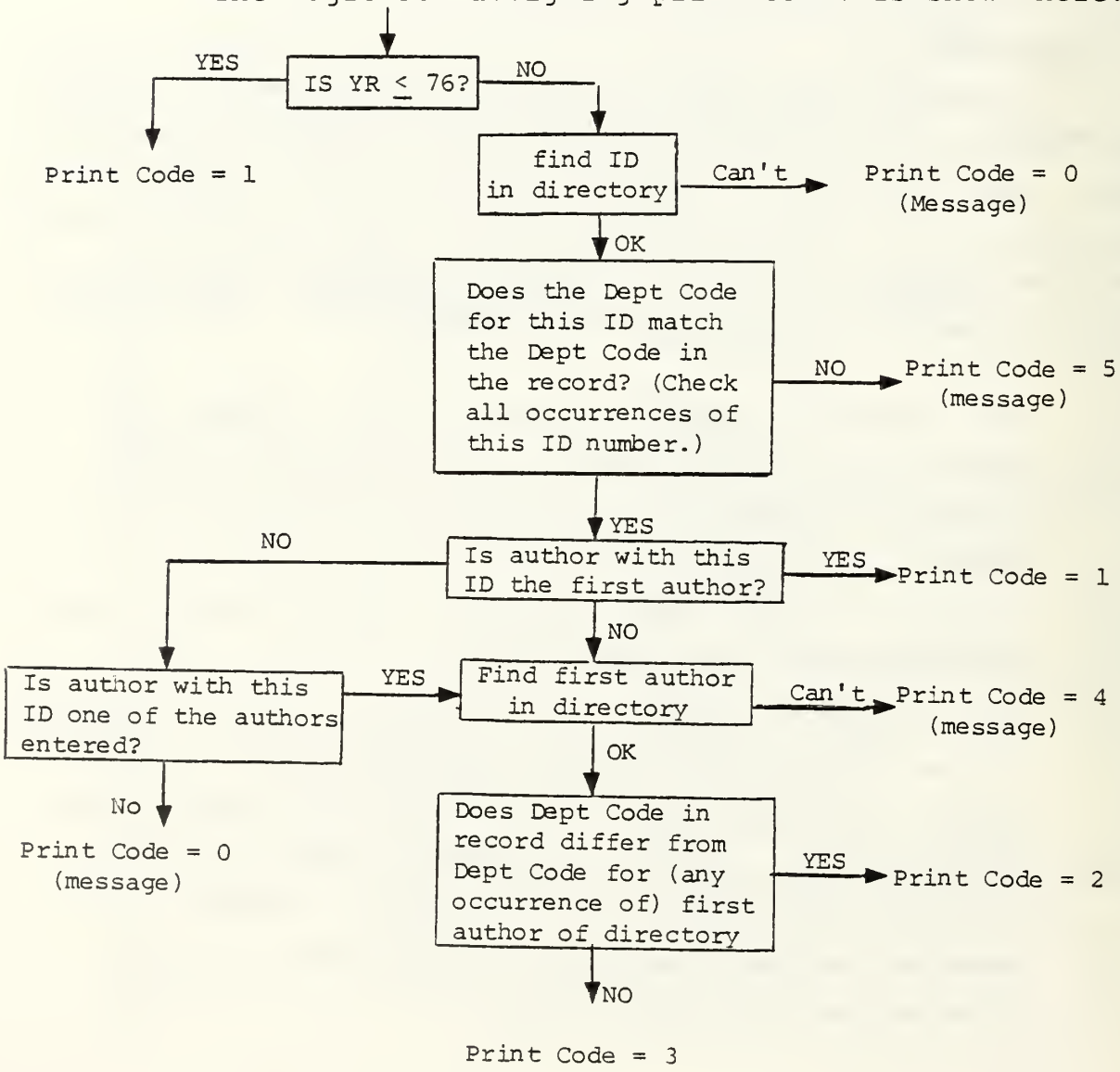
2 The faculty ID number of this record does not correspond to the first author and the department code for this author is not the same as the department code for the first author. (This is a secondary listing but in a different department from the primary listing.)

3 Same as 2 except the primary listing has the same department code as this record.

4 The first author is not in the faculty directory (he may be outside NPS, or the faculty directory may be incomplete).

5 The department code for the author with this ID number does not correspond to the department code in the record. (This normally indicates an error since we expect each faculty member's entries to bear his own department code. The error may be in the record, or the faculty member may have changed departments and the directory should be updated.)

The logic for assigning print codes is shown here.



Because of the difficulty in "backdating" the directory for past years, all new entries dated 1976 or earlier will be assigned print code = 1. Print codes 0 and 5 should not occur. If they do, they should be viewed as an indication that something is wrong, probably with the faculty directory.

In a complete listing of publications sorted by department and alphabetically by author only records containing a print code 1, 2, or 4 should be printed. This will suppress duplicate entries in the same department.

4. Faculty ID number

A unique ID number associated with the faculty member for whom this record is being entered. The list of faculty members and their ID numbers is maintained as the faculty directory. For new faculty any number may be used providing it does not conflict with an existing faculty ID number. Section III.E contains a complete list of faculty and ID numbers.

5. Security code

A single character identifies the security classification of the document.

T = Top Secret

S = Secret

C = Confidential

U = Unclassified

The INPUT program expects this in upper case.

6. Year of publication

This will be the year of publication, presentation, submission, patent application, or award, etc. The INPUT program will accept two or four digits, i.e. 79 or 1979. It will record only the 79 in either case.

7. Number of authors

This field is self-explanatory, maximum of eight authors.

8. Author number 1

This field should be typed as follows:

Howard, G T

Do not put punctuation after the initials and do not put punctuation at the end. Do include the comma after the last name followed by a space. Separate the initials by a space. (These are library requirements, the record will accept any string of 30 characters.)

9-15. Authors 2 through 8.

As above.

16. Title

Type with punctuation exactly as desired. (The printing programs will reformat the information for outputting.) Up to 240 characters are allowed. The information is accepted exactly as typed at the terminal. For very short titles the information can be typed on one line, but the program expects two, so it is

necessary to hit the carriage return a second time. For very long titles do not hit the carriage return until the remaining information is less than 120 characters in length. For titles of intermediate length, say 160 characters, the operator might type 50, or 60, or more characters (at least 40 but fewer than 120). Then hit the carriage return and continue with the remaining information.

17. Citation

Type as desired. The instructions given for the title field apply here also. A special requirement exists in the case of a book chapter (CHB) or a symposium proceedings (SYC or SYI). The library requires that the information be identified in the output as follows:

IN: ------

To accomplish this the input should include an asterisk just before the information which follows "IN." The word "IN" should not be typed. The output programs will recognize the asterisk and replace it with IN in the format shown above.

18. Translated authors

This field is of no concern to the user of the system. The INPUT program does not request information for this field. It is filled automatically by the INPUT program. It contains a 64 character translation of the author fields. The author fields are stripped of punctuation and rewritten here for later use in sorting the records alphabetically by author.

B. File Description

The records described in Section III-A are constructed using INPUT, CORRECT and DISPLAY programs described in Section III-C. They are stored in CMS files under user-selected names.

The INPUT program asks "what file"; and if no file with that name exists, it is created. New entries are written into the file in sequential order and the INPUT program reports the sequence number to the user at the terminal.

Each file is closed by the INPUT program using the CMS command FINIS after each record is written. In the event of computer failure, user error, or a communication problem only the current record will be lost.

The file size is limited only by the amount of private disk space available and this can easily be ascertained by issuing the CMS command s (for status). Likewise the current files residing on the disk can be determined with the CMS command l (for list). At any time the user should find there all previously constructed files unless they have been erased or unless there has been a disk failure. As protection against these possibilities, the files should occasionally be written on a tape.

The files are MEMO files containing records 800 characters in length. The file names can be changed using the CMS command ALTER. They can also be COMBINED, SPLIT, etc., but the records should be manipulated using only the APL programs INPUT, CORRECT, and DISPLAY. The user should not attempt to EDIT the records.

C. Program Descriptions

This section describes the programs required to use the FACPUB system. They are divided into three main categories. The program listings are found in Section V (arranged in the same order as here).

1. Programs Used at the Terminal

a. Programs used during file creation

INPUT:

Purpose--This program works interactively with the user to enter new records into the FACPUB system.

*Access: This is an APL program in the APL workspace called TT. The user must be signed on to CMS and must have entered the APL system by typing [apl] without the brackets. Then the user must issue the command, [*load tt]. The above requirements are explained in more detail in the user's guide, Section VI. Once these preliminaries have been completed, it is only necessary to type [input].*

Description--The INPUT program asks the user for the information already described under "record contents." The first question asked is "What file?", and the user may respond with any filename acceptable to CMS. It is suggested that the file names be selected to aid the user in recalling their contents. For example DEPT5279 might be the file for department 52 for the year 1979. Or the user might wish to have a file called ALL79, or ADD78. After the user has responded with a file name, INPUT will ask for the department code and will give a message which says "enter z to stop." If the user does not stop, INPUT will proceed with the questions and when one record is complete, it will return to ask for the next department code. Each item will be written into the file originally identified. To create a new file, exit from INPUT by typing z for the department code and then typing [input] again.

The INPUT program will create a file with the designated name if one does not already exist. If it does exist, the current record will be written in the file following the last record. A sequence number (row number, record number) is assigned to each entry as it is made. This number is printed at the terminal for the user's reference. It is used in the CORRECT program to locate this record if corrections to it are necessary, and it is used in the DISPLAY program. (The sequence number of each record is also displayed when printing the files using the LIST program.)

Occasionally the user will attempt to input information before the computer is ready to accept it. If this happens the message "interrupt" will appear and the current record will be lost. The user should simply begin again with that record.

It is not necessary for the user to SAVE or FILE his work (except when working with the faculty directory) since this is handled automatically by the programs. When the user has completed a session with INPUT or CORRECT or DISPLAY, he may simply sign off and his work will be retained. Likewise in the event of a computer failure (except the loss of the disk) all records except the one currently being input will be retained.

CORRECT:

Purpose--To work interactively with the user to correct existing records.

Access--(when in the APL environment with the workspace TT loaded), type [correct].

Description--After the user types [correct], the program will ask "What file?". After this question is answered, the program will ask "sequence number." The user should respond with the sequence number

(row number, record number) referred to in the description of INPUT. Next he will be asked "field number." The user should respond with a field number between 1 and 17 as described in Section III-A (Record Formats). The programs will cause the current contents of that field to be displayed and will ask the user to type the desired new contents or to hit the return key if the current contents should be retained. Then the question "field number" will be repeated.

To exit from CORRECT, respond with [0] (zero) when asked for the field number. Then the question "sequence number" will be repeated. The user can enter a new sequence number if he wishes to correct another record in the same file or he may enter [0] (zero) to exit from CORRECT.

A special message will appear at the terminal if the user enters 16 or 17 when asked for the field number. These fields are very long, 240 and 243 characters, and it is unwieldly to have to retype the entire field to correct a small error. In retyping, new errors are likely to be introduced. The special message indicates that the user can hit the return key to keep the current contents, can type [XXX] to erase the entire field, or can type three characters to identify the location of an error. In the last case the terminal will display the field from the beginning until the three characters are encountered and will ask the user to retype the remainder of the field.

DISPLAY:

Purpose--to display at the terminal the contents of any record (or records).

Access--(when in the APL environment with the workspace TT loaded), type [display].

Description--The program will ask "What file?". After this question is answered, the user will be asked to identify the sequence number(s) (row number(s), record number(s)) of the record(s) to be displayed. He should respond with a list of record numbers separated by spaces. This will cause these records to be displayed.

b. Programs used to maintain the faculty directory

ADDFAC:

Purpose--to add faculty with their ID number and department code to the faculty directory.

Access--(when in the APL environment with the workspace TT loaded) type [addfac].

Description--The ADDFAC program is self-documented. It will instruct the user to make a new entry into the directory in the following format:

[ID space space NAME space space DEPT] .

It will then display the entry and print a message "if OK issue save command, if not reload TT."

The program is modifying an APL variable AA which contains the complete faculty directory. In order for the change to be effective at the next session, it must be saved. This is done by typing the symbol for APL commands and the word save, [save] or [) save].*

If several incorrect entries are made before a correct one, and then the save command is issued, the errors will be saved too. To prevent this, if an error is made when dealing with the directory, it is wise to reload TT by typing [load TT] or [) load TT].*

FIXFAC:

Purpose--to make corrections to existing directory entries or to delete them entirely.

Access--(when in the APL environment with the workspace TT loaded) type [fixfac].

Description--The program is self-documented. It will ask the user to identify the entry needing correction by typing the ID number. This number should be unique and the program starts at the top of the directory looking for it. The program then asks if the entry is to be deleted or changed and gives instructions to do it. When completed, the program will remind the user to issue the save load tt command. (For explanation, read the description of ADDFAC.)

SEEFAC:

Purpose--to display the faculty directory.

Access--(when in the APL environment with the workspace TT loaded), type [seefac].

Description--The program simply prints the APL variable AA which contains the faculty directory. The number of lines printed is equal to the number of entries in the directory.

c. Programs used to write the files on tape

LIBTAPE:

Purpose--the files constructed by the programs INPUT and CORRECT are not suitable for printing since they contain special APL characters not recognizable by the printer. LIBTAPE will translate these symbols to the intended form and will write the files on tape.

Access--This program is to be used from the terminal when a tape is attached as device 181 with a ring. For example, suppose that a memo file called ALL79 is on the P disk. The following commands will write this file onto tape (suppose the tape has been positioned):

```
.....................
[filedef 01 dsk all79 memo]
[$ libtape]
```

Output will appear at the terminal indicating the number of records processed.

Description--It is important to understand that the file just written onto tape is a derivative of the file originally constructed by the INPUT and CORRECT programs. It is for printing only and it can not be used as the parent copy for permanent retention. Corrections to these records and additional input must be done in the original (parent) file.

The file just written onto tape is not in TAPE DUMP format and the tape command LOAD will not work on this tape. The LIBTAPE program writes an end of file (eof) after each file. If several files are to be written sequentially using LIBTAPE, the series of commands above is repeated with the name of each file substituted for ALL 79.

The disk contains a compiled (TEXT) version of LIBTAPE. A FORTRAN listing is contained in this report.

ICHAR: *One other program, ICHAR, is called as a subroutine. A text version is on the disk.*

2. APL Programs Called as Subroutines by the Programs in 1

APLNAME

This APL program accepts user supplied input to construct a CMS filename.

FILESIZE

Examines a specified CMS file to determine the number of records it contains.

WRITE

Writes an APL variable into a specified CMS file.

WRITEERROR

Identifies errors occurring when executing the WRITE program and returns a message to the user.

READ

Reads the content of a CMS file and assigns it to an APL variable.

READERROR

Returns a message to the user if an error occurs in the READ program.

RINDEX

Check the user input "pub code" and finds the corresponding single character code.

LIT

Translates a numerical value to a literal.

NUM

Translates a literal to a numerical value.

NAMECHG

Performs a translation on the author fields removing punctuation. The output is stored in the record for later used in sorting alphabetically by author.

BLANK2

Examines a variable until 2 consecutive blanks are found. It returns the portion of the variable preceding the blanks. It is called from FACULTY.

TRANS41AJ

This program is required when using the IBM2741 terminal. It is used to translate the variable before writing in the file. INPUT, CORRECT and DISPLAY must be modified for the 2741 terminal.

TRANSAJ41

Used before reading (see TRANS41AJ).

PRNTCODE

Automatically accesses the faculty directory and assigns the print code.

3. FORTRAN Programs Running Under OS Used to Produce Final Output

LIST

Purpose--This is the basic printing program for the FACPUB system. It is used to provide a sequential listing of any file after it has been processed by the program LIBTAPE.

Description--The attached program listing is set up to print files 11, 12, and 13 on tape LIB202. The input data cards specify the number of files to be read and the number of records to print from each of the files. In this case the files contained 537, 476, and 614 records respectively. The contents of LIB202 is given later in Figure 14.1.

The LIST program does no sorting on the input files; it simply prints them sequentially. For reference a sequence number is printed in the left margin. This sequence number is not part of the record but is appended by the LIST program.

LIST formats the records as shown in the following example:

```
200      55 UNP 3   9 U  78
              Marshall, K T; Richards, F R
                 Joint ORSA/TIMS index.
                    Operations Research Society and The Institut
                    Management Science, 1978.
```

The first line contains fields 1 through 6. The remaining fields are the authors, the title, and the citation. Field 18 (translated

authors) is not printed. Notice that field 2 has been translated by LIST to the three character publication code. The record contains a one-character version of the publication code.

The details of LIST can be obtained from the program listing, but it will not break words in the middle and it will paginate the output. The output is suitable for immediate reproduction or compilation into bound form although the user may wish to suppress printing of line 1.

Recall from the description of INPUT that in the case of publication codes CHB, SYC, and SYI, an asterisk is to be entered in the record in place of the word "IN." In this case, the LIST programs will output the record in the following format:

```
54 SYC 1 176 U  78

      Elster, R S
         Summary of the conference
            Marine Corps and Navy Man and Woman-Power:
            Requirements and Considerations, Leesburg, Va., Mar.
            7, 1978.
               IN Proc., Washington, D. C., Smithsonian Inst.,
                  1978, p. 16.
```

When printing the final copy use 1010 on the job card to specify upper and lower case output. See detailed examples in Section IVB.

A modification of the LIST program to cause it to begin a new page each time a new author ID is encountered is explained on the listing.

GETSORT

Purpose--This program is used to extract selected files from a master tape, merge them, and sort the resulting file as specified.

Description--The program listing shown selects files 11, 12, and 13 from tape LIB202 and merges them into a single file.

The output from this program is put in file 1 on tape LIB999. It can be printed from there using the LIST program with the correct JCL.

As with the programs LIST, BOOK, and GETBOOK, this program operates on files which have been processed by LIBTAPE.

BOOK

Purpose--The BOOK program prints the publication entitled "Recent NPS Publications." It requires an input tape containing the proper information already sorted. The program GETBOOK prepares the input tape.

Description--The listing shown assumes that the input file is available as record 1 on tape LIB999.

The document printed by the BOOK program is divided into four main categories each containing several publication codes as shown below.

a. Contributions to books

i) BK
ii) TBK
iii) CHB
iv) SYE

b. Conference presentations

i) PRC
ii) PRI

c. Journal publications and conference proceedings

i) PP
ii) PI
iii) SYC
iv) SYI
v) ABC
vi) ABI

d. Technical reports and notes

i) RPT

ii) TN

Within these categories (a, b, c, d) the records must be sorted by deparment code. Any further sorting is optional. (The listing shown for GETBOOK also sorts alphabetically by author and by year.)

The BOOK program prints the contents of the specified file in a format suitable for immediate reproduction. It begins a new page each time a new department code is encountered and it puts a new page header each time a new category of publication (as defined under a, b, c, and d above) occurs. The pages are labelled with the name of the department providing it is one of the academic departments 52, 53, 54, 55, 56, 61, 62, 63, 64, 67, 68 or 69. Other department codes, such as 012, will appear on separate pages with no department heading printed.

When printing the final copy use 1010 on the job card to specify upper and lower case output.

GETBOOK

Purpose--This program will extract specified records from the files and sort them in preparation for using the BOOK program.

Description--The program shown will examine records from tape LIB202 in files 11, 12 and 13. It will extract records and put them in groups as defined by the data cards. In this case there are 4 groups, 4 types of publications in the first group, namely those with publication codes, BK, TBK, CHB, and SYE. There are 6 in the next group, etc.

As each record is read, the program determines if it belongs in group a, b, c, d or none of these. If none, the record is bypassed. Otherwise the publication code is changed to A, B, C or D. The records are then sorted as required by the BOOK program and written on the tape LIB999 as file number 1. The contents of tape LIB202 are not altered.

D. Special Variables Used in the APL Programs

This section describes several special variables used in the APL programs. Included is a description of the use of each variable and a print out of its value.

FIM

This variable is used as an index to the data fields. It shows the starting location (minus one) for each field and the field length.

```
  0     3     4     5     8     9    12    13    43    73   103   133   163
223   253   493
  3     1     1     3     1     3     1    30    30    30    30    30    30
 30   240   243
```

PCM

This variable is a list of the APL characters corresponding to the three character publication codes.

```
⍎⍕⍟
⊥'
~⊥'
⍟⍴∩
⍟⍴⍳
⌈⍎∩
⌈⍎⍳
⌈⍎∈
⎕~⍴
~⍕
⍟⍟
⍟⍳
⍺⊥∩
⍺⊥⍳
∩⍙⊥
⍴⍟⍴
⍴⍟~
⍟⍺~
⍴∈∪
~⍴
~⍙∈
⌊⍳⌈
```

PCV

This variable is also used in assigning the publication code. It contains a list of the APL characters corresponding to the single character publication codes.

αⅠ∩⌊∊_∇∆⍳∘'⎕¦⊤○⋆?⍴⌈~↓∪

SECV

The APL characters corresponding to the four acceptable security codes are stored in this variable.

↓∩⌈~

LCV

This variable holds the APL character corresponding to the lower case values for each character of the alphabet. It is used in the function NAMECHG.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

UCV

This variable holds the upper case value for each character of the alphabet. It is also used in the function NAMECHG.

⍺⍳∩⌊∈_∇Δ⍳∘'⍞|⍤⍟?⍴⌈~+∪⍵⊃↑⊂+>,.

AA

This variable holds the faculty directory. See the next section.

NFAC

This is the number of entries in the faculty directory AA. Each entry occupies 20 positions in AA. The variable NFAC is used in the programs ADDFAC, FIXFAC and SEEFAC.

E. Faculty Directory

The current faculty directory is shown below. It is stored as the APL variable AA. The format is:

FIDNAME.....ADC

where

FID = faculty ID
(3 characters, right justified)

NAME... = name
(14 characters, left justified)

ADC = academic department code
(3 characters, right justified).

The faculty ID and the academic department code are right justified to match the contents of fields 4 and 1 in the records.

The directory below has been retyped for readability with a different spacing than that actually contained in the variable AA.

FACULTY DIRECTORY

137	Tolles, W M	12
135	Johnson, J R	35
46	Barksdale, G L	52
91	Bradley, G H	52
181	Burkhead, F	52
191	Hamming, R W	52
51	Kodres, U R	52
138	Matula, D W	52
52	Raetz, G M	52
48	Rahe, G A	52
49	Roland, R J	52
180	Schell, R R	52
50	Schneidewind,	52
53	Comstock, C	53
54	Davis, D L	53
55	Franke, R H	53
172	Jayachandran,	53
132	Kildall, G A	53
173	Kovach, D	53
192	Marks, H B	53
56	Russak, I B	53
57	Schoenstadt, A	53
59	Trahan, D H	53
187	Wang, P C C	53
58	Weir, M D	53
60	Wilde, C O	53
20	Arima, J K	54
156	Creighton, J W	54
21	Derr, C B	54
176	Elster, R S	54
41	Eoyang, C K	54
16	Fremgen, J M	54
22	Giaugue, W C	54
23	Haga, W J	54
24	Jones, C R	54
17	Judson, R R	54
18	Kline, M R	54
129	Lanson	54
123	Liao, S S	54
121	McMasters, A W	54
122	Paringer, L	54
25	Senger, J D	54
26	Weitzman, R A	54
19	Whipple, D R	54
124	Wright, C A	54
92	Andrus, A F	55
15	Barr, D R	55
47	Brown, G G	55
94	Butterworth, R	55
125	Esary, J D	55
95	Forrest, R N	55
96	Gaver, D P	55
97	Hartman, J K	55
98	Howard, G T	55
126	Jacobs, P A	55
1	Larson, H J	55
2	Lewis, P A W	55
127	Lindsay, G F	55
3	Marshall, K T	55
4	Milch, P R	55
5	Neil, D E	55
6	Parry, S H	55
128	Pilnick, S E	55
7	Poock, G K	55
8	Raike, W M	55
154	Read, R R	55
9	Richards, F R	55
10	Shubert, B O	55
11	Shudde, R H	55
12	Sovereign, M G	55
13	Taylor, J G	55
14	Thomas, M U	55
130	Tysver, J B	55
155	Washburn, A R	55
131	Zehna, P W	55
193	Burke, D P	56
29	Daniel, D C	56
27	Magnus, R H	56
194	Schutz B M	56
195	Sherwin, B M	56
204	Stolfi, R H	56
28	Valenta, J	56
203	Buskirk, F R	61
164	Cooper, A W	61
190	Cooper, J N	61
99	Coppens, A B	61
107	Crittenden, E	61
100	Fairall, C W	61
101	Harrison, D E	61
165	Kelly, R L	61
136	Kinney, G F	61
167	Medwin, H	61

FACULTY DIRECTORY Cont.

166	Milne, E A	61
177	Moose, P H	61
102	Novarini, J C	61
163	Pitthan, R	61
169	Reese, W	61
103	Reinhardt, R A	61
170	Rodeback, G W	61
104	Sanders, J V	61
105	Schacher, G E	61
168	Schwirzke, F R	61
106	Wilson, O B	61
148	Adler, R W	62
157	Baycura, O M	62
36	Burton, R W	62
147	Chan, S G	62
43	Duffin, J H	62
146	Gerba, J A	62
37	Hoisington, D	62
133	Holl, S T	62
38	Kirk, D E	62
44	Knorr, J B	62
158	Myers, G A	62
39	Ohlson, J E	62
45	Panholzer, R	62
40	Parker, S R	62
30	Powers, J P	62
48	Rahe	62
145	Rothauge, C H	62
31	Sackman, G L	62
32	Stentz, D A	62
33	Strum, R D	62
42	Tao, T F	62
34	Thaler, G J	62
35	Titus, H A	62
159	Wilcox, M L	62
185	Wozencraft, J	62
73	Chang C P	63
74	Davidson, K L	63
75	Elsberry, T L	63
76	Haltiner, G J	63
77	Haney, R L	63
199	Lau, K M W	63
78	Renard, R J	63
179	Van Der Bijl,	63
79	Williams, R T	63
189	Blandin, J S	64
93	Blandin, S W	64
142	Boynton, R E	64
161	Brennan, J P	64
162	Doran, E J	64
141	Frederiksen, P	64
140	Plotkin, N	64
109	Pluta, J E	64
110	Rittenoure, R	64
183	Saunders, R E	64
134	Sohlberg	64
184	Whitney, G A	64
186	Young, R D	64
111	Ball, R E	67
150	Bank	67
112	Biblarz, O	67
151	Gawain, T H	67
152	Kahr, C H	67
113	Layton, D M	67
114	Lindsey, G H	67
115	Miller, J A	67
116	Netzer, D W	67
117	Platzer, M F	67
149	Schmidt, L V	67
118	Shreeve. R P	67
119	Simmons, J M	67
153	Zucker, R D	67
65	Andrews, R S	68
61	Bourke, R H	68
182	Chace, A B	68
66	Denner, W W	68
67	Garwood, R W	68
62	Haderlie, E C	68
68	Jung G H	68
69	Leipper, D F	68
63	Paquette, R G	68
70	Thompson, W C	68
71	Thornton, E B	68
64	Traganza, E D	68
143	Tucker, S P	68
144	Von Schwind, J	68
72	Wickham, J B	68

FACULTY DIRECTORY Cont.

160	Boone, D H	69
86	Brock, J E	69
171	Cantin, G	69
200	Cooper, T E	69
80	Fuhs, A E	69
139	Garrison, C J	69
81	Houlihan, T M	69
87	Kelleher, M D	69
88	Marto, P J	69
82	McNelley, T R	69
89	Newton, R E	69
201	Nguyen, D H	69
83	Nunn, R H	69
84	Perkins, A J	69
178	Salinas, D	69
90	Sarpkaya, T	69
85	Vanderplaats,	69

Faculty with joint appointments

91	Bradley, G H	55
47	Brown, G G	52
156	Creighton, J W	55
120	Elster, R S	55
50	Schneidewind,	54
137	Tolles, W M	61
187	Wang P C C	56

Faculty with joint appointments will be listed in the directory for each department, but they will have the same ID number for each listing so that sorting by faculty ID number will put all the citations for that faculty member together.

F. The Input Terminal

If the IBM 2741 terminal is used for the input or correction of records, a modification is required in the APL programs INPUT, CORRECT and DISPLAY. It is necesary to translate the record before it is written into the file and to translate it back whenever it is read at the terminal. Two very brief APL programs are available for this purpose. They are

1. TRANS41AJ, used when writing from the 2741.
2. TRANSAJ41, used when reading at the 2741.

These two programs must be used if the correct character set is to be obtained.

Any ASCII data terminal can be used with the programs as described in this report. If there is any doubt a dummy record containing all the upper and lower case letters, the numbers and all special characters needed should be input. It should then be written on tape using the LIBTAPE program and printed using LIST to complete the test.

IV. DETAILED OPERATING INSTRUCTIONS (USER'S GUIDE)

A. The Input Process

1. Signing on to CMS

The best source of this information is the NPS user's guide. The brief information here is for completeness only.

a) Turn on the terminal

b) Establish communication. If by phone, dial 3025 and attach the receiver to the acoustic coupler.

c) Type *break* or *attention*.

d) Type $L_{\Lambda}1182pYY$

where

XXXX = user number

YY = terminal number

Λ = space

e) Type AAAABBBB

where

AAAA = project number (use 0525)

BBBB = cost code (use 1423)

In addition to its responses between the above steps, the terminal will end this sequence with a message indicating that CMS has been entered. To confirm this, a carriage return can be typed and the response should be "cms."

At this point the s (status) and l (list) commands can be issued to see how much disk space is available and what files currently exist. Other CMS commands can also be executed at this time. (Do not EDIT the files.)

2. Entering APL

a) When in CMS, type *apl* then a carriage return. The system should respond with:

A*P*L\N*P*S

LIBRARY DOCUMENTATION SYSTEM...*LOAD... .

If any symbol other than * appears before LOAD, type that symbol followed by *off cms*. This should return the user to CMS. Repeat Step 2.

b) If APL has been entered successfully, the type element should move over 7 spaces before coming to rest. To confirm that the user is in APL he can hit a carriage return and the type element should again move over 7 spaces.

c) Before the programs described in this document can be used, the work space called tt must be loaded. This is done by typing * *load tt*.

d) The user can now type INPUT or CORRECT or DISPLAY as desired. These program are self-explantory. After exiting from any of these programs (into APL), the user can re-enter any of them by typing the appropriate program name.

e) After loading the work space TT (step c) the directory maintenance programs ADDFAC, FIXFAC or SEEFAC can be used.

3. Signing Off or Returning to CMS

a) To sign off (from APL) type * *off*. Wait until the terminal responds then turn it off and hang up the phone.

b) If the user wishes to return to CMS (from APL) he should type * *off cms*. After the terminal responds, the user may wish to confirm that CMS has been entered. The CMS commands can be issued, APL can be re-entered, or the user can sign off by typing *cp log*.

c) One other environment might be entered inadvertently, namely CP. If a carriage return produces the response *cp*, the user should type *i cms* to re-enter CMS.

B. Printing Reports (including examples)

Once a master tape (LIBTAPE version) of the files has been constructed, sorting the files and printing final copies of the output is a fairly straightforward process.

This section will assume the existence of a master tape LIB202 whose contents are as shown in Figure IV.1.

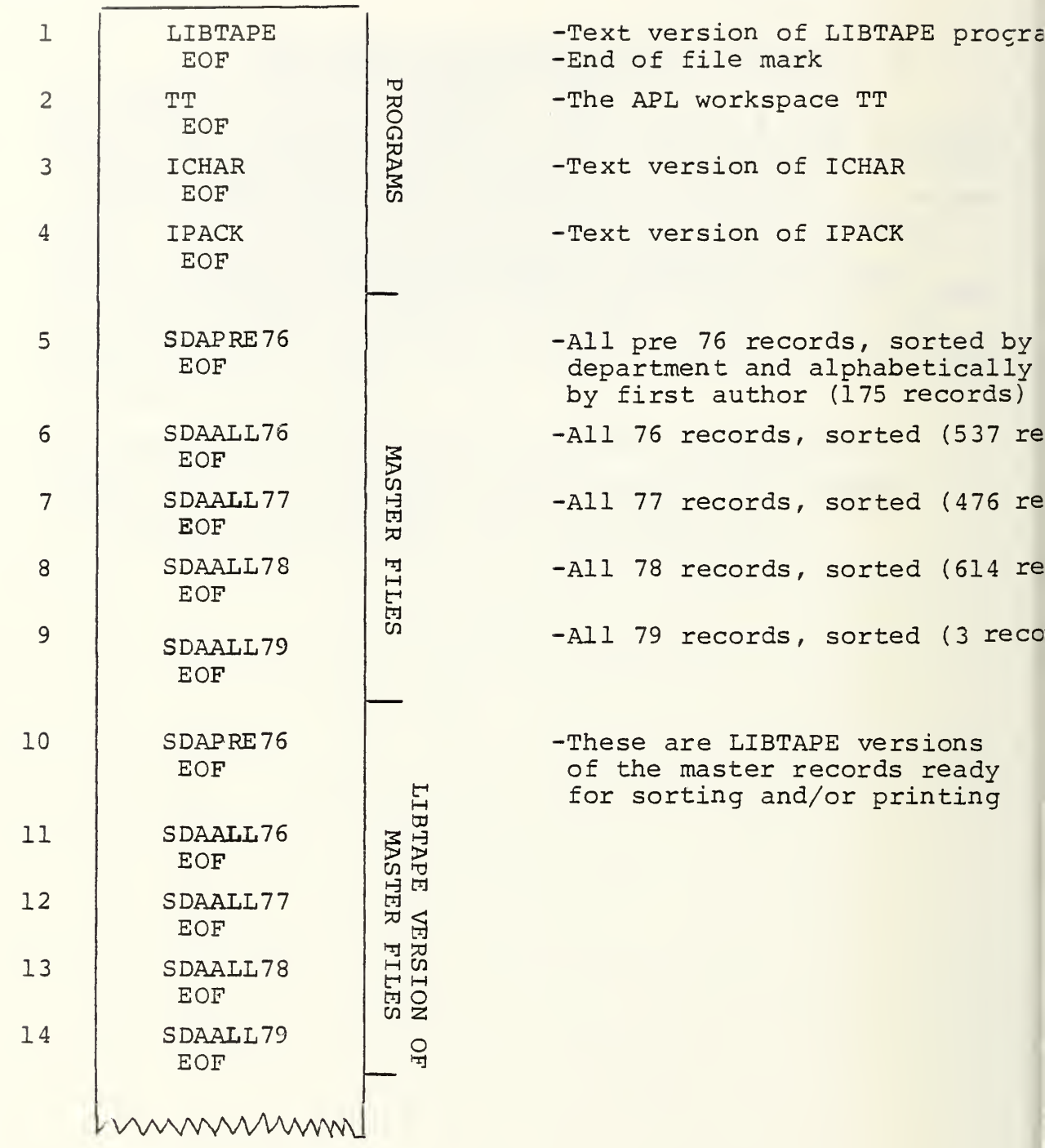

FILE
Tape LIB202
Comments
1
LIBTAPE
EOF
-Text version of LIBTAPE progra
-End of file mark
2
TT
EOF
-The APL workspace TT
3
ICHAR
EOF
-Text version of ICHAR
4
IPACK
EOF
-Text version of IPACK
PROGRAMS
5
SDAPRE76
EOF
-All pre 76 records, sorted by department and alphabetically by first author (175 records)
6
SDAALL76
EOF
-All 76 records, sorted (537 re
7
SDAALL77
EOF
-All 77 records, sorted (476 re
8
SDAALL78
EOF
-All 78 records, sorted (614 re
9
SDAALL79
EOF
-All 79 records, sorted (3 reco
MASTER FILES
10
SDAPRE76
EOF
-These are LIBTAPE versions of the master records ready for sorting and/or printing
11
SDAALL76
EOF
12
SDAALL77
EOF
13
SDAALL78
EOF
14
SDAALL79
EOF
LIBTAPE VERSION OF MASTER FILES

Figure IV.1

1. Example 1.

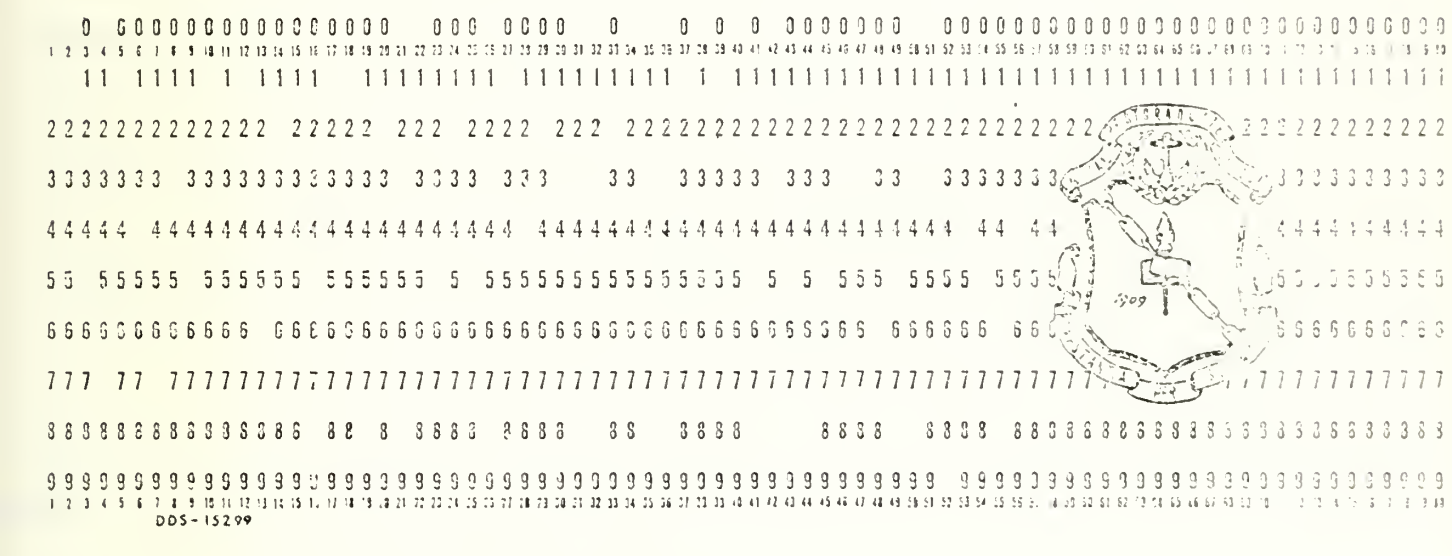
To print (the LIBTAPE version of) the files contained on tape LIB202, the following steps are required.

a) Be sure that the tape is available to the computer center. If it is stored there, it is available. If it is stored in the library, it can be delivered over the counter in the computer center.

b) Submit the LIST program through the card reader in the computer center.

i) The job card should have the following form

```
//EXAMPLE1 JOB (1182,0525,1423,,20,,1010),'HELLO',TIME=4
```



The interpretation of this card is

EXAMPLE1 = user selected name (title on printout)

JOB = identifies "job" card

1182 = user number

0525 = project number

1423 = mail code

20 = this calls for a maximum of 20 thousand lines of output. If this is too small, the job will not finish printing. This estimate is generous for this job.

1010 = this calls for overnight printing with the TN chain (upper and lower case). The job will not be available until the next morning. The job will be available behind the counter in the computer center. This field can be deleted to get an "immediate" print out with upper case only.

Hello = user selected identification to appear on printout.

Time 4 = estimate of time required in minutes. Four is a generous estimate.

i) The JCL cards should have the following form. These 15 cards will replace the first 9 JCL cards shown in the program listing.

FIRST CARD

```
//GO.FT08F001 DD DISP=(OLD,KEEP),UNIT=3400-4,DSN=ADD,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(10,NL,,IN),VOL=(,RETAIN,SER=LIB202)
               DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(11,NL,,IN),VOL=(,RETAIN,SER=LIB203)
               DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(12,NL,,IN),VOL=(,RETAIN,SER=LIB202)
               DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(13,NL,,IN),VOL=(,RETAIN,SER=LIB202)
               DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(14,NL,,IN),VOL=(,RETAIN,SER=LIB202)
```

LAST CARD

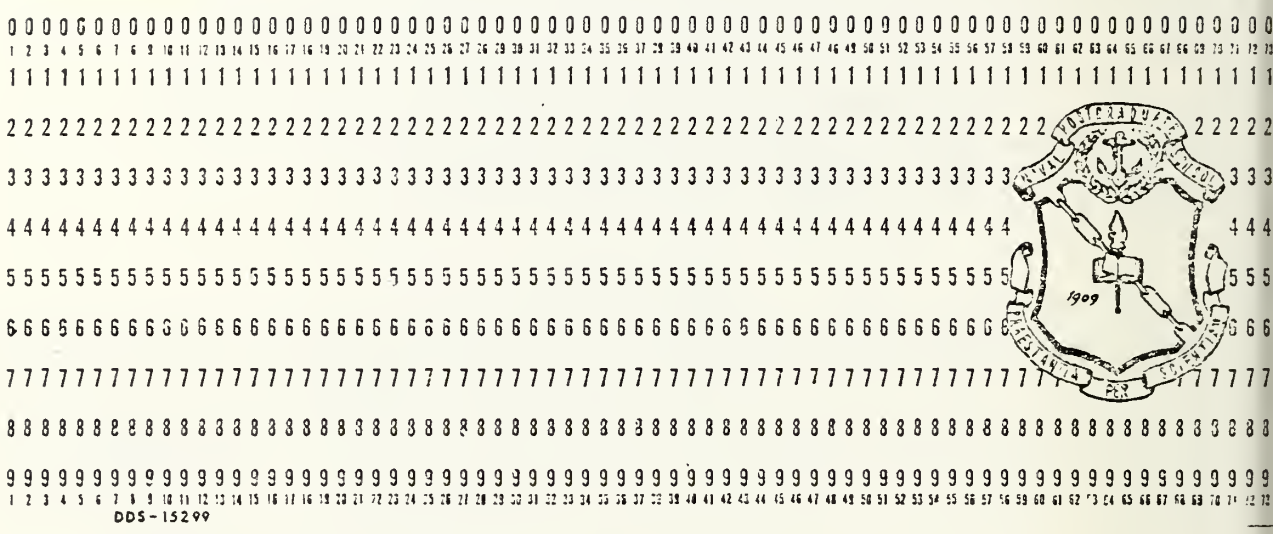
DDS-15299

iii) The data cards should be

FIRST CARD
5

175 537 476 614 .3

LAST CARD



These will replace the two data cards shown in the program listing for LIST.

The final /* card (orange card) is not shown on the listing. It should be added before submitting the job.

The job setup just described will produce a printout of the files 10 through 14 from tape LIB202. The records in each file will be numbered sequentially.

The tape LIB202 will not be changed.

2. Example 2

This example assumes that the user wants to select files 11 (SDAALL76), 12 (SDAALL77) and 13 (SDAALL78) from tape LIB202, merge them, sort them according to department code, then publication code, then year and print the results in a single sequential listing.

The following steps are required:

a) Be sure tape LIB202 is available to the computer center (see Example 1).

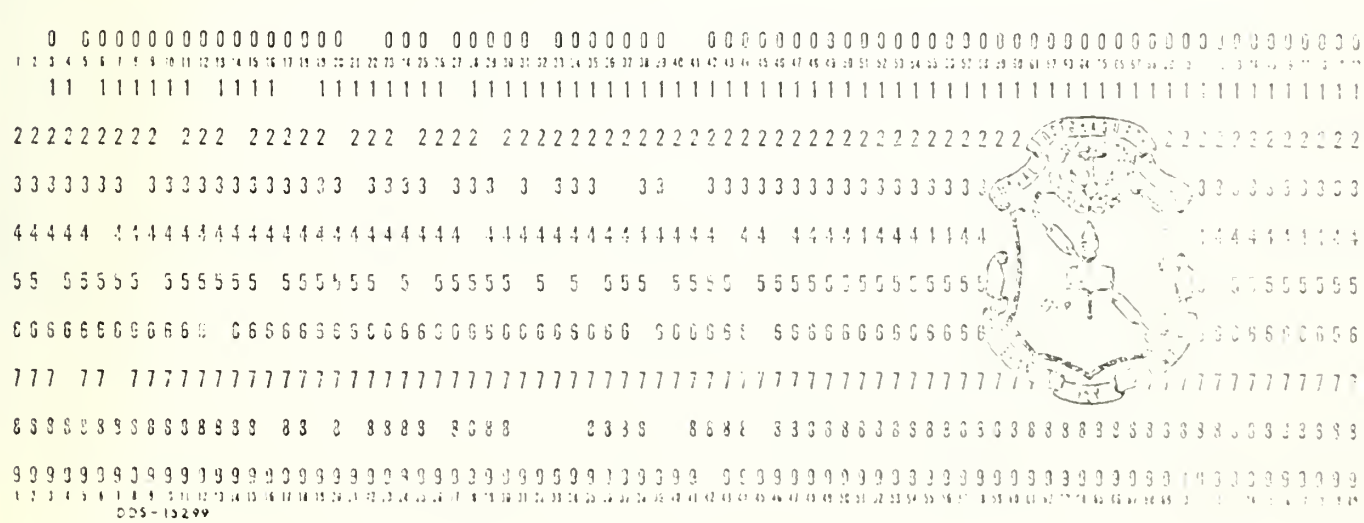
b) Select a "scratch tape" and make it available to the computer center. This example assumes tape LIB999 is used.

c) Two jobs will have to be submitted, separated in time. The first job using GETSORT will prepare the information on tape LIB999. When this is completed (it should take anywhere from a few minutes to an hour or two depending on how busy the computer center is), the second job using LIST should be submitted.

1. USING GETSORT (refer to listing)

i) Job card

```
//EXAMPLE2 JOB (1182,0585,1483),'HELLO',TIME=4
```



See example 1 for interpretation. No lengthy output is required, no printing is needed, thus the 20 and the 1010 have been deleted from the job card.

ii) JCL

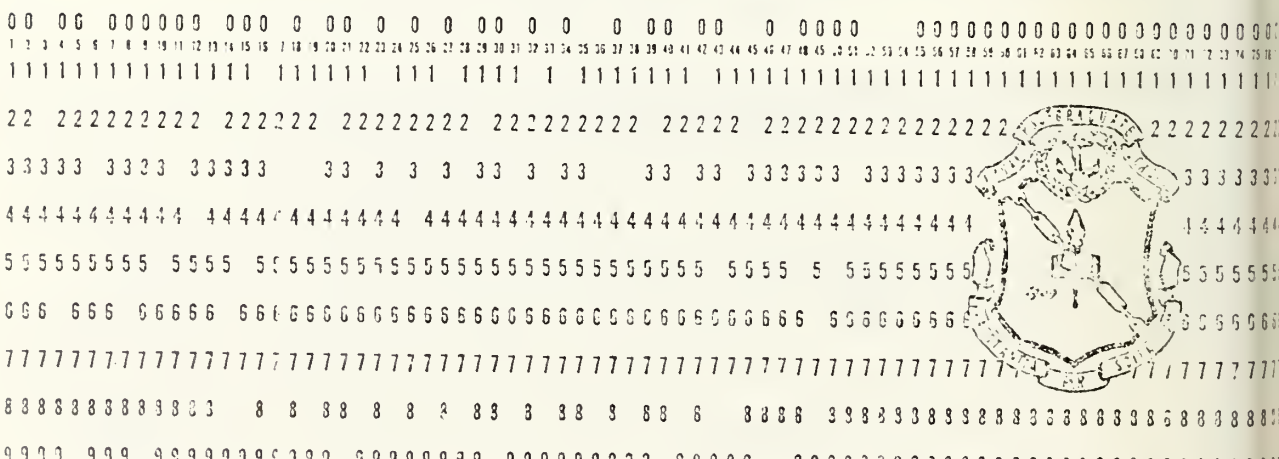
The JCL cards shown on the GETSORT listing already point to files 11, 12, and 13 (cards 24 through 32) on tape LIB202. These JCL cards are in groups of 3 and other groups can be added or some of these can be removed to point to a different file.

The third card from the end in the GETSORT listing points the output to file 1 on tape LIB999. It is correct for this example.

iii) Sort fields

The last card shown should be changed for this example to

```
SORT FIELDS=(1,3,BI,A,4,1,BI,A,10,3,BI,A),SIZE=E3000
```



The final /* card (orange card) is not shown in the listing. It should be put on the end of the deck before submitting.

The output received from this job will include a brief statement indicating that 1627 records were processed.

2. USING LIST

i) Job card (see Example 1)

10 (thousand) lines of output should be adequate.
1010 should be used on the job card to get the upper and lower case output
time = 4 is sufficient

ii) JCL (see Example 1)

The input file is now on tape LIB999 in file number 1. Modify the JCL in LIST accordingly.

iii) The data cards in LIST should indicate 1 file, 1627 records.

The /* card (orange card) completes the deck.

C. File Management

The FACPUB system can be viewed as having three major divisions:

1. The input process consisting of terminal sessions using the programs INPUT, CORRECT, DISPLAY.

2. The file management process consisting of transferring the disk files to tape.

3. The output process consisting of using LIST, GETSORT (and annually, GETBOOK and BOOK).

The first and the third of these have been described in detail already, but nothing has been said about the second. This section will deal exclusively with file management and will provide a complete plan for the near term. The user should attempt to familiarize himself with the problems involved and not merely follow thse suggestions blindly.

Several general issues will be addressed before a detaile discussion of "how to do it" is undertaken in Section D. These issues are

i) record keeping (knowing what files are available and where they are)

ii) file protection (against loss)

iii) updating and correction of files

iv) recommended file storage.

i) Record Keeping

Anyone experienced with computer use is aware of the difficulty of keeping track of exactly what files he has and what is in them, particularly if they are not used frequently.

In the FACPUB system the permanent records are stored on tape and often the user will find himself with several similar versions of a file. For example one may be sorted one way, another sorted another way. It is imperative that accurate and up-to-date records be kept regarding the file contents and location. Each tape used in the FACPUB system should be catalogued and its "index" (a written record) updated everytime it is used.

ii) File Protection

Considerable effort is involved in constructing and correcting a large file and care must be taken to be sure that no files are lost.

Files on the disk are subject to loss in the event of disk failure or inadvertant erasure. It is wise to provide back-up protection against loss by occasionally writing new files onto tape. This can easily be overdone, but perhaps each time one hundred records are entered they should be backed up with a tape copy. A single scratch tape can be used for this purpose and overwritten with a new copy of the files each time it is used.

Files which have been corrected and stored on tape for permanent retention should also be duplicated to prevent loss. This duplication must be done carefully to prevent losing the master copy in the process. The user should always be in a position where his files can be recovered if he loses all copies currently in use. That is, during a session at the terminal the user may have a tape mounted which contains valuable files

from which he intends to make a copy. He may also have these files on the disk, but he should be aware of the possibility of losing both copies because of equipment failure or user error.

Another caution which deserves emphasis in the FACPUB system is the distinction between the master copies of the files and the LIBTAPE copies. The master copy is the one constructed by INPUT and CORRECT but is not the one used to print output. The LIBTAPE version is used for that purpose. The LIBTAPE version can always be reconstructed from the master but not vice versa.

iii) Updating and Correction

In the FACPUB system the decision has been made to store records in files according to the year of publication. (See the contents of tape LIB202 for example in Section IVB.) It also seems convenient to have these records sorted in some order and they are arranged by Department Code and then alphabetically by first author. These are viewed as the master files and a print-out of the LIBTAPE version should always be available in the library. As errors are discovered in these files they should be noted on the master copy. They can all be corrected in a single session when the decision is made to do so. The sequence number in the master copy corresponds to the number in the file so that the CORRECT program can access the proper records. When the user plans to correct all errors, say in the file SDAALL77, he can load the master copy from the tape back to the disk and correct as usual. The difficulty comes in rewriting the master copy back to tape. It is not safe to rewrite the file back on the same tape in the same file simply because the tape may not be positioned precisely and the following files might be destroyed. This can be overcome if the following files are also loaded on the disk and rewritten

on the tape following the one corrected. The difficulty is that the disk may not be large enough to hold all the required files. If not, temporary disk space can be used unless it too is inadequate.

The procedure for adding records to an existing file on the tape can be handled as just described. The addition would be available on the disk (filed by year) and would be merged (using the COMBINE command) with the file loaded from the tape. Then the merged file would be sorted and replaced onto the tape, again remembering that this will potentially destroy the following files. Another caution is that when the merged file is sorted a new file, which also occupies disk space, is created. Before sorting and after assuring that the COMBINE command was executed properly, the user might want to erase the two files previously combined into the merged file.

iv) Recommended File Storage

As mentioned in the previous section the decision has been made to store the files by year (see Section IIIB for the contents of tape LIB202).

It is recommended that the same system be continued, but it is also suggested that the number of master files on a single tape not exceed 3. The reason for this comes from the difficulty in manipulating too many records even with temporary disk space. Currently the tape LIB202 holds five master files including SDAPRE76 and SDAALL79. This is manageable for now because the pre76 file is relatively small (because it is incomplete) and the 79 file has only three records. As the pre76 file grows, it is recommended that the structure be changed to that shown below in Figure IV.2. Notice that each tape in this recommendation includes all programs, a set of master files, and the corresponding LIBTAPE versions.

TAPE 1	ALL PROGRAMS	
	SDAPRE76	Master Version
	SDAPRE76	LIBTAPE Version
TAPE 2	ALL PROGRAMS	
	SDAALL76	Master Version
	SDAALL77	Master Version
	SDAALL78	Master Version
	SDAALL76	LIBTAPE Version
	SDAALL77	LIBTAPE Version
	SDAALL78	LIBTAPE Version
TAPE 3	ALL PROGRAMS	
	SDAALL79	Master Version
	SDAALL 80	Master Version
	SDAALL 81	Master Version
	SDAALL 79	LIBTAPE Version
	SDAALL 80	LIBTAPE Version
	SDAALL 81	LIBTAPE Version
TAPE 4	AS ABOVE	

Figure IV.2. Recommended Tape Storage Structure for FACPUB System.

The exact contents of each tape should follow the example of tape LIB202 which includes EOF (end of file) marks between each file. This is repeated in Figure IV.3.

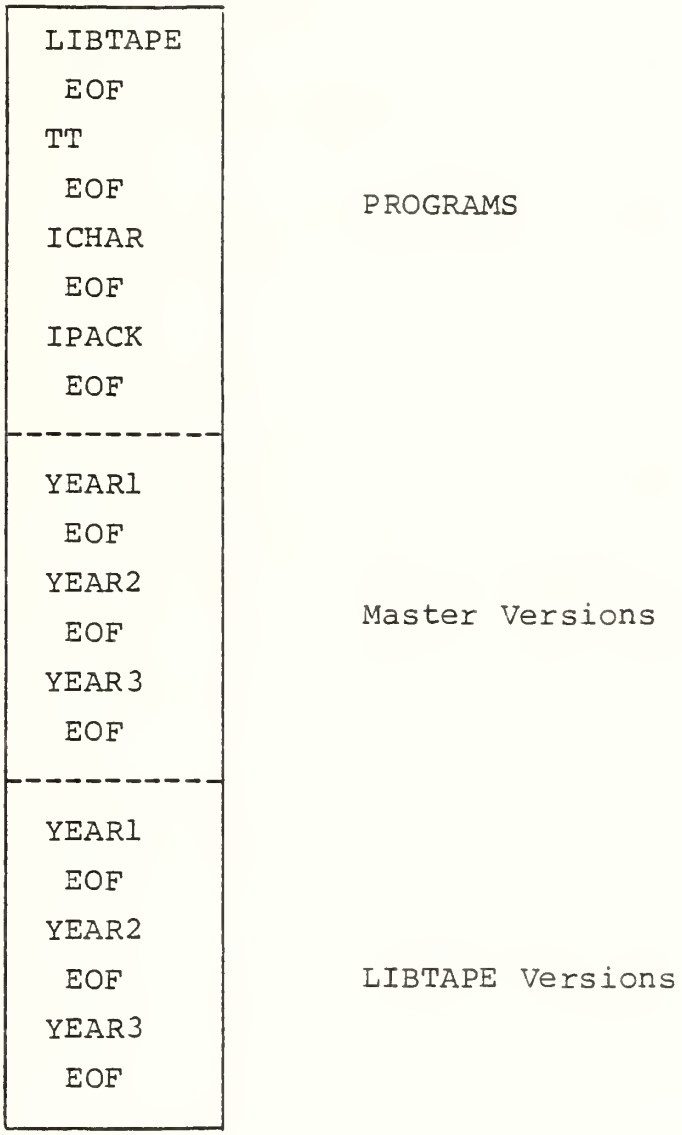


Figure IV.3. Recommended Format of Master Tapes

The reason for including the programs on each tape is that if the tape contents is loaded into temporary disk space, all the programs are readily available for use.

The reason for including the LIBTAPE version is simply that it should be stored somewhere and it can easily be printed to determine the exact contents of the master file.

If the structure recommended above is used, the process of correcting and updating files will be eased because it should always be possible to accommodate all data from any tape in the 20 cylinders of temporary disk space available.

LIST can be used to print the contents of any LIBTAPE file. The programs GETSORT and GETBOOK may sometimes have to use JCL referring to two different tapes, but the description and use of those programs is unchanged. The JCL must be changed slightly when the input comes from more than one tape. Suppose that GETSORT required the use of two tapes, say LIB202 and LIB203. Suppose that we want files 11 and 12 from LIB202 and file 5 from LIB203. In this case the JCL (see Listing of GETSORT) would be changed as follows:

```
LABEL=(12,NL,,IN),VOL=SER=LIB202
```

would replace

```
LABEL=(12,NL,,IN),VOL=(,RETAIN,SER=LIB202)
```

and

```
LABEL=(5,NL,,IN),VOL=(,RETAIN,SER=LIB203)
```

would replace

```
LABEL=(13,NL,,IN),VOL=(,RETAIN,SER=LIB202).
```

D. File Management Examples

The following topics are covered in this section:

1. How to transfer a file from disk to tape so that a copy can be obtained for proofreading.

2. How to make corrections to a file not currently on the disk when the 10 cylinders normally available are adequate.

3. How to make corrections to a file not currently on the disk when temporary space is needed.

4. How to construct a master tape as described in Section C.

5. How to add records to an existing file on tape.

1. Transferring a file to tape

Assume the file called ALL99 is on the disk and a printout is required.

a. Select a scratch tape and make it available to the computer center. This example assumes LIB999 is the tape.

b. Sign on to CMS.

c. Type

```
cp msg cp please attach tape LIB999 with ring as 181
```

Wait until the message returns that device 181 has been attached. This may take several minutes if the computer center is busy.

d. Type

```
tape rewind
```

e) Type

```
tape dump all99 memo p1
```

This will dump the master file onto the tape. It is probably a good idea to do so. It will serve as backup for the disk file and will correspond exactly to the LIBTAPE version you will be reading.

Type

```
tape writeof
```

This will write an EOF mark on the tape.

f) Type

filedef 01 dsk all99 memo

(wait for response, a second or two)

Type

$ libtape

The LIBTAPE program will execute. The number of records processed will be printed at the terminal. Wait until execution ends before touching the keyboard. This may take several minutes for a large file. Normally it will take a few seconds, maybe a minute for 50 records. The LIBTAPE program automatically writes EOF.

g) Type

tape rewind

Type

cp detach 181

Type

cp msg cp done with tape LIB999

h) Type

cp log (if you want to sign off)

The desired file is now file 2 on tape LIB999. It can be printed using LIST with the appropriate JCL.

2. Correcting a file not currently on disk

Assume we want to correct records in the files currently on tape LIB202 (see Section IVB for contents of this tape).

We will assume here that the normal space assigned for the FACPUB system is adequate to hold the necessary file.

a) Sign on to CMS

b) Type

cp msg cp please attach tape LIB202 with ring as 181

Wait until the tape is available. Availability can be confirmed by typing the next command.

c) Type

tape rewind

d) Type

tape skip 4

This will skip the first 4 files on the tape.

e) Type

tape load 5

This will load the next 5 files onto the disk. (Actually the 10 cylinders available are not sufficient to hold all this.) The tape can be rewound and detached and the operator notified. This would be the procedure if the corrections are lengthy. Alternatively, the corrections can be made immediately and the corrected version put back on the tape In either case we will assume that the files are now corrected and properly sorted.

f) Type

```
tape rewind
tape skip 4
```

g)
```
tape dump SDAPRE76 memo
tape writeof
tape dump SDAALL76 memo
tape writeof
```

.....

At this point the corrected master files are rewritten on the tape.

h) Type

```
filedef 01 dsk SDAPRE76 memo
$ libtape
(wait until execution ends)
filedef 01 dsk SDAALL76 memo
$ libtape
(wait for completion)
```

.....

Repeat these steps for each file. At that point the LIBTAPE versions are also on the tape.

i) Type

```
tape rewind
cp detach 181
cp msg cp done with tape LIB999.
```

j) Type

```
cp log
```

Since the 10 cylinders assigned to the FACPUB system are not sufficient to hold all of the LIB202, the above procedure will fail. However the process of loading selected files onto the disk is still the same. The files can then be corrected whenever time permits and they can later be rewritten onto the tape using temporary disk space (see Example 3 which follows).

3. Correcting a file on tape using temporary disk space

We will now address the case in which the 10 cylinders available for the FACPUB system are inadequate to hold the file currently on tape.

a) Sign on to CMS

b) Type

```
cp def t2314 192 20
format t all
release 191 p
release 192 t
login 192 p
login 192 b
```

At this point 20 cylinders of temporary space are available and they have been redesignated as the p disk. Be careful when typing the above commands that you do not type "format p all" as this will erase the p disk if you elect to continue. At that point in the sequence of commands the p disk is the 10 cylinders normally available.

After these commands are complete, the user might wish to type s (status) and later l (list) to confirm his position.

c) Get tape LIB202 attached and rewound as in previous examples.

d) Type

```
tape load 9
```

This will load all programs and master files from the tape onto the newly available 20 cylinders. Corrections must be made and the corrections rewritten on tape before signing off since the 20 cylinders will be lost when the user signs off. The 10 cylinders (currently designated as the b disk will then revert to the p disk).

Assuming all corrections are completed,

e) Type

```
t rewind
t skip 4
t dump SDAPRE76 memo
t writeof
t dump SDAALL76 memo
t writeof
.....................
```

f) Continue as in Step g, example 2.

4. Constructing a master tape as described in Section C.

Assume all the programs and the relevant files are on the p disk.

a) Sign on to CMS, get the desired tape attached and rewound (see Examples 1 and 2).

b) Type

```
tape dump libtape text pl
tape writeof
tape dump tt aplws pl
tape writeof
.....repeat for ICHAR, IPACK) .....
```

c) Continue as in Step g, Example 2.

5. Adding records to an existing master file

Assume temporary disk space is required as in Example 2, and assume that a new file called ADD78 is on the p-disk and must be added to the current file SDAALL78 or LIB202.

a) Sign on CMS with temporary space as in Example 3. Get the tape loaded and rewound.

b) Type

```
tape load 9
```

This will load the programs and the files SDAPRE76, SDAALL76, SDAALL77, SDAALL78, and SDAALL79.

c) Type

```
t rewind
```

The file ADD78 originally on the p-disk is on the disk now designated as b. To get a copy onto the p-disk the COMBINE command can be used. In this example the file will be combined in one step with SDAALL78.

d) Type

```
combine TOTAL78 memo p1 SDAALL78 memo p1 ADD78 memo b1
```

The file TOTAL78 should include both SDAALL78 and the ADD78 files. Confirm that it is the correct size by listing the files (l for list).

e) Type

```
erase SDAALL78 memo
```

f) Type

```
sort TOTAL78 memo SDAALL78 memo
```

The terminal will request that sort field definition be entered. Assume we want to sort by department and alphabetically by first author.

g) Type

```
1 3 737 800
```

Now the file SDAALL78 memo should contain the ADD78 file sorted as described.

All that remains is to rewrite it on tape.

h) Type

```
tape skip 7
tape dump SDAALL78 memo
tape writeof
tape dump SDAALL78 memo
tape writeof
```

Continue as in Step h, Example 2.

E. Miscellaneous Commands

The instructions discussed in this section are not intended as a tutorial. They are recorded here for completeness and should not be attempted by the user without knowledge of the CMS system or without reference to the CMS user's guide.

1. To attach a tape (with ring)

```
cp msg cp please attach tape ......
                    with ring as 181
```

2. To detach a tape

```
t  rewind
cp detach 181
cp msg cp done with tape ......
```

3. To position or scan the tape

```
t  rewind
t  skip  n  (if desired)
t  scan  n  (if desired)
t  rewind   (if desired)
```

4. To dump a file onto tape

```
t  dump filename filetype
t  writeof  (if desired)
t  rewind   (if desired)
```

5. To load the contents of a tape onto the P disk

```
( position the tape)
t  load  n
t  rewind   (if desired)
```

6. To write a file onto tape using LIBTAPE (this is in preparation for printing)

(position the tape)

```
filedef 01 dsk filename filetype
$ libtape
```

The filedef and libtape commands can be repeated for other files. Do not write an end of file since the libtape program already does this.

7. To sign on with temporary disk space

```
cp  def  t2314  192  20
format  t  all
release  191  p
release  192  t
login  192  p
login  191  b
```

8. To delete records from a file (this will change the sequence number of all following records)

The following example will erase record number 52 out of the file called ALL79. It assumes there are 53 or more records in the file.

```
split  all79  memo  XXX  memo  1  51
split  all79  memo  XXX  memo  53
erase  all79  memo
alter  XXX    memo  p1  all79  memo  p1
```

9. To transfer files to the spooled reader (or to another user number)

   cp xfer d to nnnnp (nnnn = user number)

   disk dump filename filetype filemode

   cp xfer d off

   User number nnnn can load the file by issuing the disk load command.

10. To combine or copy files

    combine newfn newft newfm oldfn oldft oldfm ...

11. To sort files under CMS

    sort oldfn oldft newfn newft

12. To change a filename

    alter oldfn oldft oldfm newfn newft newfm

V. PROGRAM LISTINGS

```
     ∇ INPUT;V;DC;APLN;S;X;J;AV;NV;NA;TV;WF
[1]    V←4ρ' '
[2]    'WHAT FILE'
[3]    WF←⍞
[4]   LRET:'~O END, PUT ⌊EPT ∩ODE  Z  '
[5]    '⌊EPT ∩ODE'
[6]    →('Z'=1↑,X←⍞)ρ0
[7]    V[⍳3]←⍎3↑⍕DC←X
[8]    S←FILESIZE APLN←APLNAME WF,' MEMO'
[9]    'THIS ENTRY WILL BE ASSIGNED SEQUENCE NUMBER ';S+1
[10]  L1:'*UB ∩ODE'
[11]   →(0≠X←(3↑X←⍞) RINDEX PCW)ρL2
[12]   '⍳NCORRECT INPUT, CHECK CODE'
[13]   →L1
[14]  L2:V[4]←PCV[X]
[15]   V←12↑V
[16]  L3:'TO αUTHS'
[17]   →(0=X←⍞)ρ0
[18]   V←V,LIT NA←8⌊X
[19]   J←1
[20]   NV←AV←0ρ' '
[21]  L4:'αUTH +';J
[22]   AV←AV,X←30↑⍞
[23]   NV←NV,NAMECHG X
[24]   →(NA≥J←J+1)ρL4
[25]   V←V,240↑AV
[26]   '~LE'
[27]   TV←⍞
[28]   V←V,240↑TV,⍞
[29]   '∩IT'
[30]   TV←⍞
[31]   V←V,243↑TV,⍞
[32]   '_ACULTY ID'
[33]   V[5+⍳3]←⍎3↑⍎⍞
[34]   '⌊ATE'
[35]   V[9+⍳3]←⍎(2↑⍎⍞),' '
[36]  L5:'⌈EC ∩ODE'
[37]   →((X←1↑,⍞)∈SECV)ρL6
[38]   '⍳NCORRECT ∩ODE, ρEENTER'
[39]   →L5
[40]  L6:V[9]←X
[41]   V←V,64↑NV
[42]   PRNTCODE
[43]   (R←V) WRITE APLN,S←S+1
[44]   CMS 'FINIS * *'
[45]   →LRET
     ∇
```

```
    ∇ CORRECT;X;J;V;Y;IV;N;AV;Z;APLN;V1;FN;XX;D;WF
[1]    'WHAT FILE'
[2]    N←FILESIZE APLN←APLNAME⍞,' MEMO'
[3]   L1:'ωHAT ⌈EQUENCE TUMBER'
[4]    →(0=Y←⍞)ρL7
[5]    →(N≥Y)ρL2
[6]    'ιNCORRECT ⌈EQUENCE TUMBER'
[7]    →L1
[8]   L2:AV←¯64↑V←800↑R←READ APLN,Y
[9]    AV←AV,26ρ' '
[10] L5:'_IELD TO≤+'
[11]   →(0=X←⍞)ρL4
[12]   'ΩURRENT'
[13]   V[IV←,FIM[1;X]+ι,FIM[2;X]]
[14]   →(X∈ 16 17)ρL20
[15] L9:'TEW + IF CURRENT OK, JUST RETURN'
[16]   →(' '=1↑Z←⍞)ρL5
[17]   →(~X∈ 16 17)ρL8
[18] L20:'RETURN TO KEEP, OR TYPE 3 CHARACTERS WHERE ERROR BEGINS, X
 TO ERASE'
[19]   →(' '=1↑Z←⍞)ρL5
[20]   →(3≠+/('XXX'=3↑Z))ρL37
[21]   'FORMER CONTENTS OF FIELD ERASED, TYPE NEW CONTENTS'
[22]   Z←⍞
[23]   →L8
[24] L37:D←¯1
[25] L23:D←D+1
[26]   →(3=+/(3↑Z)=(3↑D↓V[IV]))ρL21
[27]   →(D<FIM[2;X]-3)ρL23
[28]   'CANT FIND THESE CHARACTERS'
[29]   →L20
[30] L21:Z←D↓V[IV]
[31]   'CURRENT.. CONTINUE FROM HERE'
[32]   Z
[33] L22:Z←Z,⍞
[34] L8:→(X≠2)ρL11
[35]   →(0≠XX←(3↑Z) RINDEX PCM)ρL10
[36]   'ιNCORRECT ιNPUT, ∩HECK ∩ODE'
[37]   →L9
[38] L10:V[IV]←PCV[XX]
[39]   →L5
[40] L11:→(~X∈ 1 4)ρL12
[41]   V[IV]←⍬,FIM[2;X]↑⍬,Z
[42]   →L5
[43] L12:→(X≠6)ρL3
[44]   V[IV]←' ',¯2↑Z
[45]   →L5
[46] L3:V[IV]←FIM[2;X]↑Z
[47]   →(~X∈ 8 9 10)ρL6
[48]   AV[(30×X-8)+ι30]←NAMECHG V[IV]
[49] L6:→L5
[50] L4:(R←800↑(¯64↓V),AV) WRITE APLN,Y
[51]   CMS 'FINIS * *'
[52]   →L1
[53] L7:CMS 'FINIS * *'
    ∇
```

```
     ∇ DISPLAY;V;A;N;I;X;K;T;FN
[1]    'WHAT FILE'
[2]    FN←⍞
[3]    'WHAT SEQUENCE NUMBERS'
[4]    V←⍞
[5]    A←APLNAME FN,' MEMO'
[6]    N←⍴V←,V
[7]    I←1
[8]   L1:X←READ A,V[I]
[9]    X←736↑X
[10]   H←X
[11]   13↑X
[12]   X←13↓X
[13]   K←0
[14]  L2:X←K←K+1
[15]   →(30=+/(30⍴' '=T←30↑X))⍴L3
[16]   X←30↓X
[17]   T
[18]  L3:→(K=8)⍴L4
[19]   →L2
[20]  L4: 6 40 ⍴240↑X←253↓H
[21]   X←240↓X
[22]   3 60 ⍴243↑X
[23]   63↑180↓X
[24]   →(N≥I←I+1)⍴L1
     ∇
```

```
     ∇ ADDFAC;R;IN;J
[1]    'TYPE NEW ENTRY AS FOLLOWS'
[2]    'ID SPACE SPACE NAME SPACE SPACE DEPT'
[3]    IN←⍬
[4]    R←BLANK2 IN
[5]    J←0
[6]   L1:→(3=+/((⍬3↑⍬R)=3↑AA[(J×20)+⍳3]))⍴L3
[7]    J←J+1
[8]    →(J=NFAC)⍴L2
[9]    →L1
[10] L3:'ID NUMBER ALREADY USED, BUT OK'
[11]   20↑(J×20)↓AA
[12] L2:AA←AA,⍬3↑⍬R
[13]   IN←((⍴R)+2)↓IN
[14]   AA←AA,14↑R←BLANK2 IN
[15]   IN←((⍴R)+2)↓IN
[16]   AA←AA,⍬3↑⍬BLANK2 IN
[17]   NFAC←NFAC+1
[18]   'YOUR ENTRY IS BELOW'
[19]   20↑(20×NFAC-1)↓AA
[20]   'ISSUE SAVE COMMAND IF OK, LOAD TT COMMAND IF NOT'
[21]   ⍴0
     ∇
```

```
    ∇ FIXFAC;X;J;Z;IN
[1]   'ENTER ID NUMBER FOR ENTRY YOU WANT TO CORRECT'
[2]   X←⍎3↑⍞
[3]   J←0
[4]  L1:→(3=+/(3↑X=3↑AA[(J×20)+⍳3]))⍴L3
[5]  L7:J←J+1
[6]   →(J=NFAC)⍴L2
[7]   →L1
[8]  L3:'ENTRY IS'
[9]   20↑AA[(J×20)+⍳20]
[10]  'IS THIS THE ENTRY YOU WANT TO FIX, YES OR NO'
[11]  →(1=+/('Y'=1↑⍞))⍴L6
[12]  →L7
[13] L6:'TYPE CORRECT ENTRY AS FOLLOWS, X TO DELETE, BLANK TO KEEP'
[14]  'ID SPACE SPACE NAME SPACE SPACE DEPT'
[15]  →(' '=1↑Z←⍞)⍴0
[16]  →('X'=1↑Z)⍴L4
[17]  AA[(J×20)+⍳3]←⍎3↑⍕R←BLANK2 Z
[18]  Z←((⍴R)+2)↓Z
[19]  AA[((J×20)+3)+⍳14]←14↑R←BLANK2 Z
[20]  Z←((⍴R)+2)↓Z
[21]  AA[((J×20)+17)+⍳3]←⍎3↑⍕BLANK2 Z
[22]  'ENTRY NOW IS'
[23]  AA[(J×20)+⍳20]
[24]  'ISSUE SAVE COMMAND IF OK, LOAD TT COMMAND IF NOT'
[25]  →L5
[26] L2:'CANT FIND THE ENTRY YOU WANT, CHECK DIRECTORY'
[27]  →L5
[28] L4:AA←(J×20)↑AA,((J+1)×20)↓AA
[29]  NFAC←NFAC-1
[30]  'ENTRY DELETED, ISSUE SAVE COMMAND IF OK'
[31] L5:→⍴0
    ∇
```

```
    ∇ SEEFAC;X
[1]   X←(⍴AA)÷20
[2]   'THE NUMBER OF ENTRIES IS '
[3]   X
[4]   (X,20)⍴AA
[5]   →⍴0
    ∇
```

```
FILE: LIBTAPE  FORTRAN  P5

      INTEGER*2 IN(800)/800*Z4040/
      INTEGER*2 AUTH(6)/ZF140,ZF240,ZF340,ZF440,ZF540,ZF640/
      INTEGER*2 ZIP/ZF040/
      N=800
      CALL DSDSET(01,N,1,N)
      CALL TAPSET(2,12,800,17,1,800)
C     CALL DSDSET(02,N,1,N)
      KOUNTI=0
      KOUNTC=0
98    READ(1,100,END=99) IN
100   FORMAT(4(20A1))
      KOUNTI=KOUNTI+1
      CALL CONVRT(IN)
C        HOW MANY AUTHORS?
      DO 1 K=1,6
      IF (IN(13) .NE. AUTH(K)) GO TO 1
      KK=K
      GO TO 2
1     CONTINUE
      WRITE(6,200) KK,KOUNTI
200   FORMAT(' BAD AUTHOR COUNT: ',I3,'  SKIPPING RECORD: ',I5)
      GO TO 98
C
C        WRITE OUT ORIGINAL CITATION
C
2     WRITE(12,100) IN
      KOUNTC=KOUNTC+1
      GO TO 98
C
99    WRITE(6,201) KOUNTI,KOUNTO
201   FORMAT(' END - RECORDS IN: ',I10,'  RECORDS OUT: ',I10)
      STOP
      END
      SUBROUTINE CONVRT (IN)
      INTEGER*2 Z(64),X(64),IN(800) ,BLANK/Z4040/
      INTEGER*4 CHAR
C
      DATA Z/Z5A40,Z7F40,Z5B40,Z5040,Z4040,Z5D40,Z4040,Z5F4C,
     *Z4C4C,Z4C40,Z4B40,Z7B40,Z4E40,Z6040,Z6440,Z4040,
     *ZE640,ZC540,ZD940,ZE34C,ZE84C,ZE440,ZC940,ZD64C,
     *Z4C4C,Z4C40,Z4040,ZD740,Z5C40,Z4C40,Z4040,Z7E40,
     *Z6140,Z7940,ZC140,ZE240,ZC440,ZC740,ZC840,ZD140,
     *ZD840,Z4040,Z6B40,Z4040,ZC640,Z7D40,ZD840,Z7A40,
     *ZE94C,ZE740,ZC340,ZE540,ZC240,ZD540,Z6F40,Z4040,
     *Z4040,ZoE40,Z4040,Z4C40,ZD240,Z6C40,Z404C,Z4040/
C
      DATA X/Z8140,Z824C,Z8340,Z8440,Z8540,Z8640,Z8740,Z8840,
     *Z894C,Z4040,Z4040,Z4040,Z4040,Z4040,Z404C,Z4040,
     *Z914C,Z9240,Z9340,Z9440,Z9540,Z9640,Z9740,Z9840,
     *Z994C,Z4C4C,Z404C,Z4C40,Z404C,Z4040,Z4040,Z4040,
     *Z4040,ZA240,ZA340,ZA440,ZA540,ZA640,ZA740,ZA840,
     *ZA94C,Z4C4C,Z404C,Z4C40,Z4C40,Z4040,Z4040,ZF040,
     *ZF140,ZF240,ZF340,ZF440,ZF540,ZF640,ZF74C,ZF840,
     *ZF94C,Z4040,Z4040,Z4040,Z4040,Z4040,Z4040,Z4040/
C
      N=800
      DO 10 L=1,N
      CHAR=ICHAR(IN(L))
      IF(CHAR.LE.64) GO TO 10
      IF(CHAR.GT.128) GO TO 11
      LL=CHAR-64
      IN(L)=Z(LL)
      GO TO 10
11    IF(CHAR.LE.192) GO TO 10
      LL=CHAR-192
      IN(L)=X(LL)
10    CONTINUE
      RETURN
      END
```

```
     ∇ FID←APLNAME A;K;REM
[1]    ⍝ REMOVE EXTRA BLANKS
[2]    A←1↓(K∨1⌽K←' '≠A)/A←' ',A,' '
[3]    ⍝ FIND END OF FILENAME
[4]    K←(A⍳' ')-⍳1
[5]    ⍝ IF ONE WORD - SYNTAX ERROR
[6]    →(K=⍴A)/ER1
[7]    ⍝ EXTRACT FILENAME
[8]    FID←8↑K↑A
[9]    ⍝ AND REMAINDER
[10]   REM←(K+1)↓A
[11]   ⍝ FIND END OF FILETYPE
[12]   K←(REM⍳' ')-⍳1
[13]   ⍝ ADD FILETYPE TO FILE
[14]   FID←FID,(8↑K↑REM)
[15]   ⍝ EXTRACT 2ND REMAINDER
[16]   REM←(K+1)↓REM
[17]   ⍝ CHECK SPECIAL MODES
[18]   →((∧/'SY'=2↑REM)∨(∧/'* '=2↑REM))/L1
[19]   ⍝ MODELETTER='P' UNLESS OTHERWISE
[20]   FID←FID,'ABCTP'['ABCT'⍳1↑REM]
[21]   ⍝ MODENO='1' UNLESS OTHERWISE
[22]   FID←FID,'0234561'['023456'⍳1↑1↓REM]
[23]   →L2
[24] L1:FID←FID,2↑REM
[25]   ⍝ RECTYPE='F' UNLESS V SPECIFIED
[26] L2:FID←FID,' ','FV'[('V'=¯1↑REM)+⍳1]
[27]   ⍝ CONVERT TO EBCDIC INTEGER
[28]   FID←2 OF FID
[29]   →0
[30] ER1:'FILETYPE MISSING'
     ∇
```

```
     ∇ N←FILESIZE FID
[1]    N←102I'QSIZE FID'
     ∇
```

```
    ∇ Y WRITE FID
[1]   101I'QWRITE FID Y'
[2]   →(0=104I0)/0
[3]   WRITEERROR 104I0
[4]   *
    ∇


    ∇ WRITEERROR EC
[1]   ⍝ SELECT MESSAGE APPROPRIATE TO ERROR CODE
[2]   →(EC= 0 1 4 5 7 12 14 15 16)/M0+ 0 1 2 2 6 3 3 5 4
[3]   ⍝ GIVE STANDARD MESSAGE FOR OBSCURE OR 'IMPOSSIBLE' MESSA
[4]   →(EC∈ 2 3 6 8 9 11 13 17 18 19)/L1
[5]   →0,ρ□←'INVALID ERROR CODE'
[6]  L1:'WRITE ERROR NO. ';EC;' - CONSULT MANUAL'
[7]   →0
[8]  M0:→0,ρ□←'WRITE OK'
[9]   →0,ρ□←'INVALID FILENAME'
[10]  →0,ρ□←'INVALID FILEMODE'
[11]  →0,ρ□←'FILE IS READ-ONLY'
[12]  →0,ρ□←'F-V ERROR'
[13]  →0,ρ□←'RECORD LENGTH ERROR'
[14]  →0,ρ□←'FILE INDEX ERROR'
    ∇


    ∇ A←READ FID
[1]   A←102I'QREAD FID'
[2]   →(0=104I0)/0
[3]   READERROR 104I0
[4]   *
    ∇


    ∇ READERROR EC
[1]   ⍝ SELECT MESSAGE APPROPRIATE TO ERROR CODE
[2]   →(EC= 0 1 12)/M0+ 0 1 2
[3]   ⍝ GIVE STANDARD MESSAGE FOR OBSCURE OR 'IMPOSSIBLE' MESSAG
[4]   →(EC∈ 2 3 4 5 6 7 8 9 11 13)/L1
[5]   →0,ρ□←'INVALID ERROR CODE'
[6]  L1:'READ ERROR NO. ';EC;' - CONSULT MANUAL'
[7]   →0
[8]  M0:→0,ρ□←'READ OK'
[9]   →0,ρ□←'FILE NOT FOUND'
[10]  →0,ρ□←'FILE INDEX ERROR'
    ∇
```

```
     ∇ R←A RINDEX M;I;J
[1]    R←((×/0=X),X←∧/M[;ιJ]=(I,J←ρ,A)ρA)/0,ιI←1↑ρM
[2]    ⍝ FINDS INDEX OF ALL ROWS OF M STARTING WITH A, A,M NUM OR LIT≥
, GIVES 0 IF A NOT IN M
     ∇



     ∇ R←LIT A
[1]    R←,'0123456789'[1+((1+⌊10⊛A)ρ10)⊤A]
     ∇



     ∇ NV←NUM V
[1]    NV←10⊥('0123456789'ιV)-1
     ∇

     .


     ∇ R←NAMECHG V;TV;T
[1]    TV←V
[2]    TV[(V∈UCV)/ιρV]←LCV[((1+ρUCV)≠T)/T←UCVιV]
[3]    R←TV
     ∇



     ∇ L←BLANK2 X;A;K;LOOP
[1]    LOOP←0
[2]    K←ρX
[3]    L←1↑X
[4]   L1:→(2=+/('  '=A←2↑X←1↓X))ρ0
[5]    LOOP←LOOP+1
[6]    L←L,1↑A
[7]    →(LOOP=K-1)ρ0
[8]    →L1
     ∇



     ∇ R←TRANS41AJ V;X;Y;A1;A2
[1]    A1←'→∧)×≠∨÷→[]¨¯<≤≥>-(;:'
[2]    A2←' ∨¯≥)≠( ≤<-;[]>: ¨  '
[3]    R←V
[4]    R[X/ιρ,V]←A2[(X←21≠Y)/Y←A1ιV]
     ∇



     ∇ R←TRANSAJ41 V;X;Y;A1;A2
[1]    A1←'×[]¨¯<≤≥>≠∨-÷→();:'
[2]    A2←' <≤()][×≥∨∧¨:≥÷≠¯>'
[3]    R←V
[4]    R[X/ιρ,V]←A2[(X←19≠Y)/Y←A1ιV]
     ∇
```

```
    ∇ PRNTCODE;ID;AUT;N;PC;M;YR;DC;FIRST;FOUND;NA;AUTH;J
[1]   FIRST←0
[2]   FOUND←0
[3]   NA←NUM 1↑12↓V
[4]   ID←3↑5↓V
[5]   AUT←14↑13↓V
[6]   DC←3↑V
[7]   YR←2↑10↓V
[8]   →((NUM YR)≤76)ρL4
[9]   N←0
[10] L3:N←N+1
[11]  →(3=+/(ID=3↑(20×N-1)↓AA))ρL2
[12]  →(N<NFAC)ρL3
[13]  →(FOUND=1)ρL9
[14]  PC←0
[15]  'ID NBR NOT IN DIRECTORY. PRINT CODE 0 ASSIGNED'
[16]  →L8
[17] L2:→(3≠+/DC=3↑(17+20×N-1)↓AA)ρL10
[18]  →(14=+/(AUT=AUTH←14↑(3+20×N-1)↓AA))ρL4
[19]  J←0
[20] L11:J←J+1
[21]  →(J=NA+1)ρL12
[22]  →(14=+/(AUTH=14↑(13+30×(J-1))↓V))ρL13
[23]  →L11
[24] L12:'FACULTY WITH THIS ID IS NOT ONE OF THE AUTHORS ENTERED'
[25]  PC←0
[26]  'PRINT CODE 0 ASSIGNED'
[27]  →L8
[28] L13:M←0
[29] L6:M←M+1
[30]  →(14=+/(AUT=14↑(3+20×M-1)↓AA))ρL5
[31]  →(M<NFAC)ρL6
[32]  →(FIRST=1)ρL8
[33]  PC←4
[34]  'FIRST AUTHOR NOT IN DIRECTORY. PRINT CODE 4 ASSIGNED'
[35]  →L8
[36] L10:→(N=NFAC)ρL9
[37]  FOUND←1
[38]  →L3
[39] L9:'DIRECTORY SHOWS THIS AUTHOR NOT IN DEPT ENTERED'
[40]  'PRINT CODE 5 ASSIGNED'
[41]  PC←5
[42]  →L8
[43] L4:PC←1
[44]  →L8
[45] L5:FIRST←1
[46]  →(3=+/(3↑(17+20×M-1)↓AA)=DC)ρL7
[47]  PC←2
[48]  →L8
[49] L7:PC←3
[50]  →L6
[51] L8:PC←1↑1↓LIT 10+PC
[52]  V[5]←PC
    ∇
```

```
//   EXEC  FORTCLG,REGION.GO=180K
//FORT.SYSIN DD *
C
C   LIST
C
C   THIS IS THE BASIC PRINTING PROGRAM FOR THE FACPUB SYSTEM
C   IT OPERATES ON FILES WHICH HAVE BEEN CONSTRUCTED USING THE APL PROGRAMS
C   INPUT AND CORRECT AND HAVE THEN BEEN TRANSLATED USING THE PROGRAM LIBTAPE
C
C   THE PROGRAM PROVIDES A SEQUENTIAL LISTING OF THE FILE CONTENTS
C   A SEQUENCE NUMBER IS ALSO PRINTED
C
C   SETUP....
C     JOB CARD, CECK, END
C     JCL
C     3 CARDS SHOWING A    UNP B    BK-.......
C     DATA CARD SHOWING NUMBER OF FILES, IS
C     DATA CARD SHOWING NUMBER IN EACH FILE, ALL IS
C     CRANGE CARC
C
C      WITH A VERY MINOR CHANGE THIS PROGRAM CAN BE USED TO
C     SKIP TO A NEW PAGE EACH TIME A NEW AUTHOR IS ENCOUNTERED
C     THIS IS EXPLAINED ON COMMENT CARDS IN THE LISTING BELOW
C     DO NOT DO THIS UNLESS THE INPUT TAPE IS SORTED BY AUTHOR ID
C     OR YOU WILL GET A NEW PAGE WHEN YOU DONT WANT IT
C
C
      DIMENSION ALPHA(2,22)
      DIMENSION ALP1(6)
      DIMENSION NIF(20)
      INTEGER * 2 IALP2(30,8), IALP3(240), IALP4(243)
      INTEGER * 2 IPL(60)
      INTEGER * 2 IBLK /'  '/, ISCO /';'/, ISTAR /'*'/, INDA /'IN'/
      INTEGER * 2 IWDA
      DATA BLK/'    '/
      AUTH=BLK
      LMX = 60
C
      LAU = 30
      LMT = 56
      LTI = 240
      LMS = 53
      LSO = 243
      IAL = 22
      LICR = 15
      LINE = 75
      LMAX = 75
      READ(5,1) ALPHA
    1 FORMAT(10(1X, A1, 3X, A3))
      IC = 0
      IA = 7
      READ(5,3) NOF
      NFC =99999
      NPF = 1
      IF (NOF .LT. 1) GO TO 10
      READ(5,3) (NIF(I), I=1,NOF)
    3 FORMAT(16I5)
      NFC = NIF(1) + 1
   10 CONTINUE
      READ(8,4,END=80) ALP1, NA, IALP2, IALP3, IALP4
    4 FORMAT(A3,A1,A1,A3,A1,A3, I1, 240A1, 240A1, 243A1)
      IC = IC + 1
      IF (IC .LT. NFC) GO TO 5
      NPF = NPF + 1
      IF (NPF .GT. NOF ) GO TO 80
      NFC = NIF(NPF) + 1
      IC = 1
    5 CONTINUE
C   TO GET A NEW PAGE FOR EACH NEW AUTHOR INSERT CARD BELOW
C   DO THIS ONLY IF THE INPUT IS SORTED BY AUTHOR
C   THERE IS ONE OTHER CARD THAT MUST BE INSERTED
C     IF(AUTH.NE.ALP1(4))GO TO 70
      IF ((LINE+LICR) .GT. LMAX) GO TO 70
    2 CONTINUE
```

```
      DO 8 I = 1,IAL
      IF (ALP1(2) .NE. ALPHA(1,I)) GO TO 8
      ALP1(2) = ALPHA(2,I)
      GO TO 9
    8 CONTINUE
    9 CONTINUE
      WRITE(6,11)
      LINE = LINE + 2
   11 FORMAT(//)
      WRITE(6,12) IC, ALP1
   12 FORMAT(1X, I4, 4X, A3,1X,A3,1X,A1,1X,A3,1X,A1,1X,A3 /)
      LINE = LINE + 2
      IBL = 1
      LC = 1
      DO 30 I = 1,NA
      IF (IBL .LT. 1) GO TO 16
      DO 15 IB = LC,LMX
   15 IPL(IB) = IBLK
      IBL = 0
   16 CONTINUE
      DO 20 II = 1,LAU
      IF (IALP2(II,I) .EQ. IBLK .AND. IALP2(II+1,I) .EQ. IBLK) GO TO 1
      GO TO 20
   18 CONTINUE
      IALP2(II,I) = ISCO
      ICH = II + 1
      LCT = LC + ICH - 1
      IF (LCT .GT. LMX) GO TO 26
      GO TO 21
   20 CONTINUE
   21 II = 0
      DO 24 J = LC,LCT
      II = II + 1
      IPL(J) = IALP2(II,I)
   24 CONTINUE
      LC = LCT + 1
      GO TO 30
   26 CONTINUE
      WRITE(6,27) IPL
   27 FORMAT(15X, 60A1)
      LINE = LINE + 1
      IBL = 1
      LC = 1
      LCT = ICH
      GO TO 21
   30 CONTINUE
      IF (IBL .LT. 1) GO TO 32
      DO 31 IB = LC, LMX
   31 IPL(IB) = IBLK
   32 CONTINUE
      IPL(LC-2) = IBLK
      WRITE(6,27) IPL
      LINE = LINE + 1
      IFN = 0
      II = 0
      J = 1
      LC = 0
      ICH = 0
   34 CONTINUE
      II = II + 1
      LC = LC + 1
      ICH = ICH + 1
      IF (IALP3(II) .NE. IBLK) GO TO 36
      IF (LC .EQ. 1) J = J + 1
      ICH = 0
      IF (IALP3(II+1) .NE. IBLK) GO TO 36
      IF (IALP3(II+2) .NE. IBLK) GO TO 36
      IFN = 1
      GO TO 38
   36 CONTINUE
      IF (LC  .GT. LMT) GO TO 38
      IF (II .LT. LTI) GO TO 34
      IFN = 1
      GO TO 33
```

```
38 CONTINUE
   II = II - ICH
33 CONTINUE
   WRITE(6,39) (IALP3(K), K=J,II)
   LINE = LINE + 1
   IF (IFN .GT. 0) GO TO 40
   ICH = 0
   J = II + 1
   LC = 0
   IF (II .LT. LTI) GO TO 34
39 FORMAT(18X, 57A1)
40 CONTINUE
   IFN = 0
   LWS = LMS
   II = 0
   IS = 0
   J = 1
   IWDA = INDA
   LC = 0
   ICH = 0
44 CONTINUE
   II= II + 1
   LC = LC + 1
   ICH = ICH + 1
   IF (IALP4(II) .NE. IBLK) GO TO 46
   IF (LC .GT. 1) GO TO 45
   J = J + 1
   LC = 0
45 CONTINUE
   ICH = 0
   IF (IALP4(II+1) .NE. IBLK) GO TO 47
   IF (IALP4(II+2) .NE. IBLK) GO TO 47
   IFN = 1
   GO TO 48
46 CONTINUE
   IF (IALP4(II) .EQ. ISTAR) GO TO 52
47 CONTINUE
   IF (LC .GT. LWS) GO TO 48
   IF (II .LT. LSO) GO TO 44
   IFN = 1
   GO TO 50
48 CONTINUE
   II = II - ICH
50 CONTINUE
   IF (IS .GT. 1) GO TO 60
   WRITE(6,49) (IALP4(K), K=J,II)
   ICH = 0
   J = II + 1
   LC = 0
   LINE = LINE + 1
   IF (IS .GT. 0) GO TO 54
   IF (IFN .GT. 0) GO TO 10
   IF (II .LT. LSO) GO TO 44
49 FORMAT(21X, 54A1)
   GO TO 10
52 IS = 1
   ICH = 0
   II = II - 1
   IF (LC .GT. 1) GO TO 48
54 IS = 2
   J = J + 1
   LWS = 47
   II = II + 1
   GO TO 44
60 CONTINUE
   WRITE(6,61) IWDA, (IALP4(K), K=J,II)
61 FORMAT(24X, A2, 1X, 48A1)
   LINE = LINE + 1
   IF (IFN .GT. 0) GO TO 10
   IWDA = IBLK
   ICH = 0
   J = II + 1
   LC = 0
   IF (II .LT. LSO) GO TO 44
```

```
      GO TO 10
   70 CONTINUE
      LINE = 0
C  TO GET A NEW PAGE FOR EACH NEW AUTHOR INSERT CARD BELOW
C  DO THIS ONLY IF THE INPUT IS SORTED BY AUTHOR
C  THERE IS ONE OTHER CARD THAT MUST BE INSERTED
C     AUTH=ALPI(4)
      WRITE(6,71)
   71 FORMAT('1')
      GO TO 2
   80 CONTINUE
      STOP
      END
//GO.FT08F001 DD DISP=(OLD,KEEP),UNIT=3400-4,DSN=ADD,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(11,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//             DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(12,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//             DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=ADD,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(13,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//GO.SYSIN DD *
 A   UNP B   BK- C   TBK D   PRC E   PRI F   SYC G   SYI H   SYE I   LTR J
 K   PP- L   PI- M   ABC N   ABI O   CHB P   RPR Q   RPT R   PAT S   REV T
 U   THE V   MIS
    3
  537  476  614
```

```
//*
//*   GETSORT
//*
//*   THIS PROGRAM IS FOR USE IN THE FACPUB SYSTEM
//*   IT IS FOR FILES THAT HAVE BEEN PROCESSED BY LIBTAPE
//*   IT IS THE PRIMARY MEANS OF EXTRACTING RECORDS FROM THE MASTER TAPE
//*
//*   THIS PROGRAM WILL READ THE CONTENTS OF THE SPECIFIED FILES
//*   IT WILL THEN SORT THE RECORDS AS INDICATED ON THE LAST CARD
//*   THE SORTED RECORDS WILL THEN BE WRITTEN ON A NEW TAPE
//*   THE LIST PROGRAM SHOULD BE USED TO PRINT THE NEW FILE
//*
//SORTEM  EXEC  PGM=IERRCOOO,REGION=150K
//SORTPR  DD   SYSOUT=A
//SORTLIB   DD   DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK02    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK03    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK04    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK05    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK06    DD   UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTIN        DD  DISP=(OLD,KEEP),UNIT=3400-4,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(11,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//                DD  DISP=(OLD,KEEP),UNIT=AFF=SORTIN,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(12,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//                DD  DISP=(OLD,KEEP),UNIT=AFF=SORTIN,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(13,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//SORTOUT  DD  DISP=(NEW,KEEP),UNIT=(3400-4,,DEFER),DSN=SORTED,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(1,BLP),VOL=SER=LIB999
//SYSIN DD *
  SORT FIELDS=(4,1,BI,A,1,3,BI,A,737,64,BI,A,10,3,BI,A),SIZE=E3000
```

```
//  EXEC  FORTCLG,REGION.GO=18CK
//FORT.SYSIN DD *
C
C   BOOK
C
C
C   THIS PROGRAM BOOK IS USED TO PRINT THE VOLUMN ENTITLED
C   "RECENT NPS PUBLICATIONS".  IT REQUIRES AN INPUT TAPE CONTAINI
C   RECORDS SORTED INTO 4 GROUPS
C      A CONTRIBUTIONS TO BOOKS
C      B JOURNAL PUBLICATIONS AND CONFERENCE PROCEEDINGS
C      C CONFERENCE PRESENTATIONS
C      D TECHNICAL REPORTS
C   THE INPUT TAPE IS CONSTRUCTED USING THE PROGRAM GETBOOK
C
C   THIS PROGRAM WILL PRINT A RECORD ONLY IF THE PRINT CODE IS 1 C
C   THUS IT WILL SUPPRESS DUPLICATES WITHIN A DEPARTMENT
C
C   IF THE DEPARTMENT CODE ON A RECORD IS NOT ONE OF THE ACADEMIC
C   CODES, THE PAGE HEADING PRINTED FOR THAT RECORD WILL CONTAIN A
C   DEPT OF ....
C
C   SETUP...
C      JOB, DECK, END
C      JCL SPECIFYING INPUT FILE
C      DATA.  PAGE HEADER FOR GROUP 1
C      DATA.  PAGE HEADER FOR GROUP 2
C      DATA.  PAGE HEADER FOR GROUP 3
C      DATA.  PAGE HEADER FOR GROUP 4
C      DATA.  END OF PUB CODES
C      DATA.  ALL
C
       DIMENSION ALP1(6)
       DATA BLK/'    '/, UC/'U'/
       DIMENSION  PCODE(23)
       REAL * 4 ENDC /'END'/
       REAL * 4 HEAD1(7) /'NAVA', 'L PC', 'STGR', 'ADUA', 'TE S',
      1 'CHOO', 'L   '/
       REAL * 4 HEAD2(5) /Z049695A3, Z85998A8, Z6E40C381, Z9389869
      1 Z99958981/
       REAL * 8 CONT /Z4D839695A37D845D/
       REAL * 4 HEAD3(4) / 'DEPA', 'RTME', 'NT C', 'F   '/
       REAL * 4 DCODE(11) /' 52 ', ' 53 ', ' 54 ', ' 55 ', ' 56 ',
      1 ' 61 ', ' 62 ', ' 63 ', ' 67 ', ' 68 ', ' 69 '/
       DIMENSION DEPT(7,12)
       REAL * 4 C52(7)/'COMP','UTER',' SCI','ENCE','    ','    ',
      1 '    '/
       REAL * 4 C53(7) /'MATH','EMAT','ICS ','    ','    ','    ',
      1 '    '/
       REAL * 4 C54(7) /'ADMI','NIST','RATI','VE S','CIEN','CES ',
      1 '    '/
       REAL * 4 C55(7) /'OPER','ATIO','NS R','ESEA','RCH ','    ',
      1 '    '/
       REAL * 4 C56(7) /'NATI','ONAL',' SEC','URIT','Y AF','FAIR',
      1 'S   '/
       REAL * 4 C61(7) /'PHYS','ICS ','AND ','CHEM','ISTR','Y   ',
      1 '    '/
       REAL * 4 C62(7) /'ELEC','TRIC','AL E','NGIN','EERI','NG  ',
      1 '    '/
       REAL C63(7) /'METE','ORCL','OGY ','    ','    ','    ',
      1 '    '/
       REAL * 4 C67(7) /'AERO','NAUT','ICS ','    ','    ','    ',
      1 '    '/
       REAL * 4 C68(7) /'OCEA','NOGR','APHY','    ','    ','    ',
      1 '    '/
       REAL * 4 C69(7) /'MECH','ANIC','AL E','NGIN','EERI','NG  ',
      1 '    '/
       REAL * 4 CBLK(7) /'    ','    ','    ','    ','    ','    ',
      1 '    '/
       EQUIVALENCE (DEPT(1,1),C52(1)), (DEPT(1,2),C53(1)),
      1   (DEPT(1,3),C54(1)), (DEPT(1,4),C55(1)), (DEPT(1,5),C56(1)
      1   (DEPT(1,6),C61(1)), (DEPT(1,7),C62(1)), (DEPT(1,8),C63(1)
      1   (DEPT(1,9),C67(1)), (DEPT(1,10),C68(1)), (DEPT(1,11),C69(
```

```
     1   (DEPT(1,12),CBLK(1))
         REAL * 8 PUB(4,23)
         DIMENSION NIF(20)
         INTEGER * 2 IALP2(30,8), IALP3(240), IALP4(243)
         INTEGER * 2 IPL(60)
         INTEGER * 2 IBLK /'  '/, ISCO /':'/, ISTAR /'*'/, INDA /'IN'/
         INTEGER * 2 IWDA
         DATA AB/'B   '/,AC/'C   '/,AO/'O   '/,AD/'D   '/
         DATA AE/'E   '/,AF/'F   '/,AG/'G   '/,AH/'H   '/
         DATA AM/'M   '/,AN/'N   '/,AK/'K   '/,AL/'L   '/
         DATA AJ/'J   '/,AQ/'Q   '/
         DATA P1/'1   '/,P2/'2   '/,P4/'4   '/,P5/'5   '/
C
         LMX = 60
C
         LAU = 30
         LMT = 56
         LTI = 240
         LMS = 53
         LSO = 243
         LEND = 7
         LENP = 4
         POK = 0.
         IDE = 11
         IAL = 5
         LICR = 10
         LMAX = 56
         LINE = 75
         CPUB = BLK
         CDEPT = BLK
         IP = 4
         IC = 0
         IA = 7
         NFC =99999
         WRITE(6,102)
  102    FORMAT('1  LIST OF PUB CODES & HEADERS' //)
         I = 0
  100    I = I + 1
         READ(5,101) PCODE(I), (PUB(J,I), J=1,IP)
  101    FORMAT(A3, 2X, 4A8)
         IF ( PCODE(I) .EQ. ENDC) GO TO 105
         WRITE(6,103) I, PCODE(I), (PUB(J,I), J=1,IP)
  103    FORMAT(7X, I2, 2X, A4, 4A8)
         GO TO 100
  105    CONTINUE
         IAL = I - 1
         READ(5,1) UU
    1    FORMAT(A1)
         ICLASS = 2
         IF (UU .EQ. UC) ICLASS= 1
   10    CONTINUE
         READ(8,4,END=80) ALP1, NA, IALP2, IALP3, IALP4
         IF(ALP1(3).EQ.P4.OR.ALP1(3).EQ.P5) GO TO 777
         IF(ALP1(3).EQ.P1.OR.ALP1(3).EQ.P2) GO TO 777
         GO TO 10
  777    CONTINUE
         IF (ICLASS .EQ. 2) GO TO 106
         IF (ALP1(5) .NE. UC) GO TO 10
  106    CONTINUE
    4    FORMAT(A3,A1,A1,A3,A1,A3, I1, 240A1, 240A1, 243A1)
         IC = IC + 1
         IF (ALP1(2) .EQ. CPUB) GO TO 6
C        NE PUB-CODE
C
         CPUB = ALP1(2)
         DO 13 I = 1,IAL
         IF (CPUB .NE. PCODE(I)) GO TO 13
         POK = 1.
         IPPU = I
         GO TO 7
   13    CONTINUE
         POK = 0.
         GO TO 10
    6    CONTINUE
```

```
      IF (POK .EQ. 0.) GO TO 10
      IF (ALP1(I) .EQ. CDEPT) GO TO 5
    7 CONTINUE
      INPUB = 0
      CDEPT = ALP1(1)
      DO 8 I = 1,IDE
      IF (CDEPT .NE. DCODE(I)) GO TO 8
      IPDE = I
      GO TO 70
    8 CONTINUE
      IPDE = IDE + 1
      GO TO 70
    5 CONTINUE
      IF ((LINE+LICR) .GT. LMAX) GO TO 70
    2 CONTINUE
    9 CONTINUE
      WRITE(6,11)
   11 FORMAT(' ')
      LINE = LINE + 1
      IBL = 1
      LC = 1
      DO 30 I = 1,NA
      IF (IBL .LT. 1) GO TO 16
      DO 15 IB = LC,LMX
   15 IPL(IB) = IBLK
      IBL = 0
   16 CONTINUE
      DO 20 II = 1,LAU
      IF (IALP2(II,I) .EQ. IBLK .AND. IALP2(II+1,I) .EQ. IBLK) GO TO
      GO TO 20
   18 CONTINUE
      IALP2(II,I) = ISCO
      ICH = II + 1
      LCT = LC + ICH - 1
      IF (LCT .GT. LMX) GO TO 26
      GO TO 21
   20 CONTINUE
   21 II = 0
      DO 24 J = LC,LCT
      II = II + 1
      IPL(J) = IALP2(II,I)
   24 CONTINUE
      LC = LCT + 1
      GO TO 30
   26 CONTINUE
      WRITE(6,27) IPL
   27 FORMAT(15X, 60A1)
      LINE = LINE + 1
      IBL = 1
      LC = 1
      LCT = ICH
      GO TO 21
   30 CONTINUE
      IF (IBL .LT. 1) GO TO 32
      DO 31 IB = LC, LMX
   31 IPL(IB) = IBLK
   32 CONTINUE
      IPL(LC-2) = IBLK
      WRITE(6,27) IPL
      LINE = LINE + 1
      IFN = 0
      II = 0
      J = 1
      LC = 0
      ICH = 0
   34 CONTINUE
      II = II + 1
      LC = LC + 1
      ICH = ICH + 1
      IF (IALP3(II) .NE. IBLK) GO TO 36
      IF (LC .EQ. 1) J = J + 1
      ICH = 0
      IF (IALP3(II+1) .NE. IBLK) GO TO 36
      IF (IALP3(II+2) .NE. IBLK) GO TO 36
```

```
      IFN = 1
      GO TO 38
   36 CONTINUE
      IF (LC  .GT. LMT) GO TO 38
      IF (II .LT. LTI) GO TO 34
      IFN=1
      GO TO 33
   38 CONTINUE
      II = II - ICH
   33 CONTINUE
      WRITE(6,39) (IALP3(K), K=J,II)
      LINE = LINE + 1
      IF (IFN .GT. 0) GO TO 40
      ICH = 0
      J = II + 1
      LC = 0
      IF (II .LT. LTI) GO TO 34
   39 FORMAT(18X, 57A1)
   40 CONTINUE
      IFN = 0
      LWS = LMS
      II = 0
      IS = 0
      J = 1
      IWDA = INDA
      LC = 0
      ICH = 0
   44 CONTINUE
      II= II + 1
      LC = LC + 1
      ICH = ICH + 1
      IF (IALP4(II) .NE. IBLK) GO TO 46
      IF (LC .GT. 1) GO TO 45
      J = J + 1
      LC = 0
   45 CONTINUE
      ICH = 0
      IF (IALP4(II+1) .NE. IBLK) GO TO 47
      IF (IALP4(II+2) .NE. IBLK) GO TO 47
      IFN = 1
      GO TO 48
   46 CONTINUE
      IF (IALP4(II) .EQ. ISTAR) GO TO 52
   47 CONTINUE
      IF (LC .GT. LWS) GO TO 48
      IF (II .LT. LSO) GO TO 44
      IFN=1
      GO TO 50
   48 CONTINUE
      II = II - ICH
   50 CONTINUE
      IF (IS .GT. 1) GO TO 60
      WRITE(6,49) (IALP4(K), K=J,II)
      ICH = 0
      J = II + 1
      LC = 0
      LINE = LINE + 1
      IF (IS .GT. 0) GO TO 54
      IF (IFN .GT. 0) GO TO 10
      IF (II .LT. LSO) GO TO 44
   49 FORMAT(21X, 54A1)
      GO TO 10
   52 IS = 1
      ICH = 0
      II = II - 1
      IF (LC .GT. 1) GO TO 48
   54 IS = 2
      J = J + 1
      LWS = 47
      II = II + 1
      GO TO 44
   60 CONTINUE
      WRITE(6,61) IWDA, (IALP4(K), K=J,II)
   61 FORMAT(24X, A2, 1X, 48A1)
```

```
      LINE = LINE + 1
      IF (IFN .GT. 0) GO TO 10
      IWDA = IBLK
      ICH = 0
      J = II + 1
      LC = 0
      IF (II .LT. LSO) GO TO 44
      GO TO 10
   70 CONTINUE
      LINE = 0
      WRITE(6,71)
   71 FORMAT('1'/'0'/'0')
      WRITE(6,72) HEAD1
   72 FORMAT(33X, 7A4)
      WRITE(6,73) HEAD2
   73 FORMAT(35X, 5A4)
      WRITE(6,74) HEAD3, (DEPT(I,IPDE),I=1,LEND)
   74 FORMAT(//30X, 3A4,A2, 7A4)
      IF (INPUB .LT. 1) GO TO 77
      WRITE(6,75) (PUB(I,IPPU),I=1,LENF), CONT
      WRITE(6,11)
   75 FORMAT(/ 30X, 4A8, 1X, A8)
      GO TO 2
   77 CONTINUE
      WRITE(6,75) (PUB(I,IPPU), I=1,LENF)
      WRITE(6,11)
      INPUB = 1
      GO TO 2
   80 CONTINUE
      WRITE(6,71)
      STOP
      END
//GO.FT08F001 DD DISP=(OLD,KEEP),UNIT=3400-4,DSN=ADD,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(1,NL,,IN),VOL=(,RETAIN,SER=LIB999)
//GO.SYSIN DD *
A     CONTRIBUTIONS TO BOOKS
B     JOURNAL PUBS & CONFERENCE PROCS
C     CONFERENCE PRESENTATIONS
D     TECHNICAL REPORTS AND NOTES
END OF PUB CODES
ALL
```

```
//  EXEC  FORTCLG
//FORT.SYSIN DD *
C
C   GETBOOK
C
C   THIS PROGRAM GETBOOK IS USED TO PREPARE A TAPE FROM WHICH THE BOOK
C   "RECENT NPS PUBLICATIONS" CAN BE PRINTED.  THE PRINTING IS DONE WITH
C   THE PROGRAM CALLED BOOK
C
C   THIS PROGRAM WILL EXTRACT RECORDS BY PUB CODE FROM THE FILES SPECIFIED,
C   GROUP THEM AS INSTRUCTED IN THE DATA, SORT THEM (SEE LAST CARD),
C   AND WRITE THEM ON A NEW TAPE READY FOR PRINTING BY BOOK.
C
C
C   SETUP
C     JOB, DECK, END
C     JCL SPECIFYING INPUT FILES...
C     THREE CARDS SHOWING A   UNP B   BK- .....
C     DATA CARD SHOWING NUMBER OF GROUPS
C     DATA CARD SHOWING NUMBER OF PUB CODES IN FIRST GROUP
C     DATA CARD SHOWING PUB CODES IN FIRST GROUP
C     REPEAT LAST TWO CARDS FOR REMAINING GROUPS....
C     15 JCL CARDS RELATED TO SORTING AND WRITING THE OUTPUT TAPE
C     THE OUTPUT TAPE AND FILE NUMBER ARE SPECIFIED ON CARD 14
C     THE FINAL CARD SPECIFIES HOW THE FILES ARE TO BE SORTED
C
      DIMENSION SELE(20,10), ISV(10)
      DIMENSION ALL(199), ALPHA(2,22), SELECT(22)
      REAL RAN(10) /'A','B','C','D','E','F','G','H','I','J'/
      IGMX = 10
      ISMX = 20
      IN = 0
      IC = 0
      IAL = 22
      READ(5,1) ALPHA
    1 FORMAT(10(1X, A1, 3X, A3))
      READ(5,2) IGRP
      IF (IGRP .LE. IGMX) GO TO 14
      WRITE(6,57) IGMX, IGRP
   57 FORMAT('1  ERROR # OF GROUPS EXCEEDS MAX OF ', I6,
     1   '  VALUE IS ', I6)
      STOP
   14 CONTINUE
      DO 20 K = 1,IGRP
      READ(5,2) IS
    2 FORMAT(I4)
      ISV(K) = IS
      IF (IS .LE. ISMX) GO TO 15
      WRITE(6,58) RAN(K), ISMX, IS
   58 FORMAT('1  ERROR # IN GROUP ', A1, '  EXCEEDS MAX OF ', I6,
     1   '  VALUE IS ', I6)
      STOP
   15 CONTINUE
      READ(5,3) (SELECT(I), I=1,IS)
    3 FORMAT(20A4)
      WRITE(6,59) RAN(K), (SELECT(J),J=1,IS)
   59 FORMAT(///'  PUB CODES IN GROUP ', A1, '  ARE: ',  / (2X, 20A5))
      DO 4 I = 1,IS
      SA = SELECT(I)
      DO 5 J = 1,IAL
      IF (SA .NE. ALPHA(2,J)) GO TO 5
    6 CONTINUE
      SELE(I,K) = ALPHA(1,J)
      GO TO 4
    5 CONTINUE
      WRITE(6,7) SA
    7 FORMAT('1  ERROR IN INPUT PUB CODE ', A4)
      STOP
    4 CONTINUE
   20 CONTINUE
    9 CONTINUE
      READ(8,10,END=50) AA, APUB, ALL
      IF (IN .GT. 50) GO TO 101
      WRITE(6,56) AA,APUB, (ALL(I), I=1,20)
```

```
  101 CONTINUE
   56 FORMAT(1X, A4, A2, 20A4)
      IN = IN + 1
      DO 13 K = 1,IGRP
      IS = ISV(K)
      DO 8 I = 1,IS
      IF (APUB .EQ. SELE(I,K)) GO TO 12
    8 CONTINUE
      GO TO 13
   12 CONTINUE
      APUB = RAN(K)
      WRITE(9,10) AA, APUB, ALL
      IC = IC + 1
      GO TO 9
   13 CONTINUE
      GO TO 9
   10 FORMAT(A3,A1,199A4)
   50 WRITE(6,51) IN, IC
   51 FORMAT('1  END OF RUN:  RECORDS IN ', I6, '  AND OUT ', I6)
      STOP
      END
//GO.FT08F001 DD DISP=(OLD,KEEP),UNIT=3400-4,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(11,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//             DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(12,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//             DD DISP=(OLD,KEEP),UNIT=AFF=FT08F001,DSN=DUM,
//  DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//    LABEL=(13,NL,,IN),VOL=(,RETAIN,SER=LIB202)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(10,1)),DSN=TC.SORT,
//    DCB=(RECFM=FB,LRECL=800,BLKSIZE=800),DISP=(NEW,PASS,DELETE)
 A    UNP B    BK- C    TBK D    PRC E    PRI F    SYC G    SYI H    SYE I    LTR J
 K    PP- L    PI- M    ABC N    ABI O    CHB P    RPR Q    RPT R    PAT S    REV T
 U    THE V    MIS
    4
    4
BK- TBK CHB SYE
    6
PP- PI- SYC SYI AEC ABI
    2
PRC PRI
    2
RPT TN-
//SORTEM  EXEC  PGM=IERRCOOO,REGION=150K
//SORTPR  DD  SYSOUT=A
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK02   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK03   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK04   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK05   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTWK06   DD  UNIT=SYSDA,SPACE=(CYL,(15),,CONTIG)
//SORTIN DD UNIT=SYSDA,DSN=TC.SORT,DISP=(OLD,DELETE,DELETE),
//   DCB=(RECFM=FB,LRECL=800,BLKSIZE=800),LABEL=(,,,IN)
//SORTOUT DD DISP=(NEW,KEEP),UNIT=(3400-4,,DEFER),DSN=SORTED,
// DCB=(RECFM=FB,LRECL=800,BLKSIZE=800,DEN=2),
//   LABEL=(1,BLP),VOL=SER=LIB999
//SYSIN DD *
  SORT FIELDS=(4,1,BI,A,1,3,BI,A,737,64,BI,A,10,3,BI,A),SIZE=E3000
```

REFERENCES

[1] Recent Naval Postgraduate School Publications, Report No. NPS-012-79-004PR, Naval Postgraduate School, Monterey, CA, July 1979.

[2] Marshall, K. T. and Richards, F. R. Joint ORSA/TIMS Index. Operations Research Society and the Institute of Management Science, 1978.

INITIAL DISTRIBUTION LIST

	NO. OF COPIES
Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	10
Library, Code 55 Naval Postgraduate School Monterey, CA 93940	1
Naval Postgraduate School Monterey, CA 93940	
Attn: G. T. Howard, Code 55Hk	10
R. J. Stampfel, Code 55	1
Fran Wheeler, Code 0141	1
Library, Code 0141 Naval Postgraduate School Monterey, CA 93940	1
W. M. Tolles Dean of Research Code 012 Naval Postgraduate School Monterey, Ca. 93940	1